nature chemical biology

Article

https://doi.org/10.1038/s41589-024-01711-4

Multicellular artificial neural network-type architectures demonstrate computational problem solving

Received: 23 June 2023

Accepted: 26 July 2024

Published online: 16 September 2024

Check for updates

Deepro Bonnerjee[1,2,3], Saswata Chakraborty[1,2,3], Biyas Mukherjee[1,2,3], Ritwika Basu[1,2], Abhishek Paul[1,2] & Sangram Bagh[1,2]

Here, we report a modular multicellular system created by mixing and matching discrete engineered bacterial cells. This system can be designed to solve multiple computational decision problems. The modular system is based on a set of engineered bacteria that are modeled as an 'artificial neurosynapse' that, in a coculture, formed a single-layer artificial neural network-type architecture that can perform computational tasks. As a demonstration, we constructed devices that function as a full subtractor and a full adder. The system is also capable of solving problems such as determining if a number between 0 and 9 is a prime number and if a letter between A and L is a vowel. Finally, we built a system that determines the maximum number of pieces of a pie that can be made for a given number of straight cuts. This work may have importance in biocomputer technology development and multicellular synthetic biology.

The ability to engineer biomolecular interactions to perform human-designed computation[1] paved the way for biocomputer technology development, with projected applications in medicine[2,3], materials[4] and biotechnology[5,6]. In vitro biological computation using DNA and enzymes demonstrated a range of computations starting from molecular logic circuits, including molecular adder[7], half adder[8], half subtractor[8] and DNA data storage[9]. In vitro biomolecular computing has also been used to build artificial neural networks (ANNs)[8] and solve computational problems like Hamiltonian path problems[10], chess problems[11] and maze problems[12]. On the other hand, adapting the engineering principle in cell biology regime allowed synthetic biologists to create genetically encoded cellular devices with living cells, which performed logical operations[13,14]. Such cellular systems worked as various logic gates[14–16], half adders[17], counters[18], DeMux and Mux[19], full adder[20,21], subtractor[21] and an analog-to-digital converter[22]. Performing computations with engineered living cells has enormous importance in biocomputer technology development at the micron scale[23–25], where microprocessor-based computers have limitations due to energy, cost and technological constraints[25]. There are many areas where cell-based computers may overperform conventional computers, including computing at the micron scale, the possibility of replicative computing machines, energy requirements and cost[25]. Thus, the cell-based computer may have unprecedented applications at the micron scale in smart therapy[26–28], bioproduction[29], bioremediation[30] and smart living materials[31].

However, the success of cell-based computing technology depends on expanding the horizons of implementing complex computational device functions in living cells. Current cell-based computing functions are focused on building various logic circuits and device functions[13–22]. Such system designs[13–15] follow the hierarchical electronic design principle, where the component genetic logic gates are layered according to the electronic analogous design either in a single cell[16–18] or in a multicellular distributed format[19,20,22]. However, layering multiple logic gates in a single cell results in cross-talk among various parts and hinders building complex functionality[20,22]. Multicellular distributed computing has been adopted to build more complex functions, where various engineered cells carry a part of the integrated circuits and are connected with other cells through

[1]Biophysics and Structural Genomics Division, Saha Institute of Nuclear Physics, Block AF, Sector-I, Bidhannagar, Kolkata, India. [2]Homi Bhabha National Institute, Training School Complex, Anushaktinagar, Mumbai, India. [3]These authors contributed equally: Deepro Bonnerjee, Saswata Chakraborty, Biyas Mukherjee. e-mail: sangram.bagh@saha.ac.in

engineered chemical communications[19,20,22]. As each engineered cell is physically insulated from the other engineered cells, this reduces the cross-talk. Such strategies were used to build integrated logic circuits[19] and maze-solving biocomputers[32] in bacteria and an adder[20] and analog-to-digital converter[22] in mammalian cells.

However, such multicellular systems require multiple layers of various engineered cells and depend on the availability of the orthogonal diffusible chemical communication channels and their proper tuning. Further, many of them are mathematically unpredictable and demonstrate a single function. One of the long-term goals of synthetic biology and bioengineering is to create configurable and programmable biological systems by just mixing and matching 'Lego'-like biomodules to build multiple predictable complex and versatile biological functions[33].

However, one of the key challenges for cell-based biocomputer development is building abstract computational problem-solving capabilities[34]. Although many abstract computational problems have been solved using in vitro biomolecular computation[10–12], most of the cellular biocomputing systems have demonstrated logic circuits and device functions[13–22]. The ability of cellular computers to address computational problems of a mathematical nature is largely missing. A few recent works demonstrated graph coloring computational problem solving by a coupled network of oscillating cardiac cells[34,35] and maze problem solving by engineered bacteria[32].

Here, we introduce a mix and match modular multicellular system where, by mixing and matching discrete engineered bacterial cells, we built multiple and versatile complex computational functions that demonstrate device functions and answer a set of mathematical problems by deciding yes/no and giving the answer in terms of a number on demand. For example, we built cellular systems that identify if a number n is a prime number, check if a letter is a vowel from a string of letters and determine the maximum number of pieces of a pie obtained from a specific number of straight cuts and 'print' the answers by expressing fluorescent proteins with specific patterns. All of them are classic computational problems and standard tasks in computer coding. In computers, such tasks are performed by compiling and changing the high-level language to machine-level language in terms of '0s' and '1s'. Solving such problems with engineered cells would be a step forward in cell-based computer technology development.

To create those functions in a modular mix and match system, we adapted the basic concepts of ANN architecture to represent engineered bacteria as artificial neurosynapses or 'bactoneurons', wherein a heterogeneous population of such cells constitutes a single-layer ANN. Adaptation of ANNs in in vitro biomolecular systems[8] demonstrated several functionalities. Previously, we demonstrated[36,37] that such single-layer ANN-type architectures could also be engineered with engineered living bacteria to generate multiple computing functions like decoders, encoders, majority functions, Fredkin gates and double Feynman gates. Later, another approach for generating ANNs with bacteria was demonstrated[38].

Motivated by the success of the ANN design concept in living cells, in this work, by mixing and matching 14 engineered cells and their output variants, we built seven single-layered bactoneural networks, which demonstrated two device functions and solved nine computational decision problems and one optimization problem.

Results

Design of the configurable subtractor and adder

First, we started building the systems to demonstrate two device functions, a full subtractor and an adder. Both functions are complex to build genetically in cells. A mammalian multicellular full adder[20] required 22 component genetic logic circuits in nine different types of cells, which were layered according to the electronic circuit design and connected through engineered cellular communications. Although multicellular adder- and subtractor-like behavior has been shown in bacteria[39], unlike chemical inducer inputs with any protein output-based synthetic genetic circuits, they respond against antibiotics, and their outputs are very specific, live and dead cells. Further, none of the mammalian and bacterial adder and subtractor systems characterized and demonstrated the digital-like threshold function from low expression to high expression of reporter genes within a narrow range of input concentration.

The subtractor and adder truth tables are mapped in a single-layer ANN-type architecture format (Fig. 1a–c and Supplementary Fig. 1) following a set of rules coherent with ANN[36]. In short, first we segregated each of the truth tables into its individual output channels and considered the relationship of that specific output with all the input combinations. Next, we grouped those combinations into smaller truth tables such that each input corresponding to a particular output possesses a weight with only one unique type of sign: '+', '−' or '0'. We now reduced the resultant truth table by ignoring any component with a weight value of 0. If the output value 1 (true) in the smaller resultant truth table appeared only once, we terminated fragmenting the truth table. Otherwise, we kept fragmenting the truth table until the above condition appeared. The nine resultant truth tables represented the functional and mathematical characteristics of nine 'artificial neurosynapses' (Fig. 1a), or unit bactoneurons, which need to be engineered as cellular devices. Together, the nine bactoneurons work as a configurable cellular system, such that when the corresponding seven-unit bactoneurons are assembled into a single-layered ANN-type architecture, both the subtractor and adder could be realized by interchanging two bactoneurons from a library of nine.

Design of cellular devices and unit bactoneuron

The bactoneurons were constructed by building transcriptional synthetic genetic circuits in the bacteria (Extended Data Fig. 1). The inputs X_1, X_2 and X_3 were replaced with three chemical inducers, *N*-acyl homoserine lactone (AHL), isopropyl β-D-1-thiogalactopyranoside (IPTG) and anhydrotetracycline (aTc), respectively. The outputs O_1 and O_2 were manifested with fluorescent proteins E2-Crimson and enhanced green fluorescent protein (EGFP), respectively (Extended Data Fig. 1). Synthetic genetic circuits inside bacteria process chemical inputs and produce output by expressing fluorescent proteins, such that the input–output relation mimics a conventional activation function[40] for an artificial neurosynapse (Eq. (1)). Thus, for a given bactoneuron j,

$$O_j = 1/[1 + e - (X_1.w_{j1} + X_2.w_{j2} + X_3.w_{j3} + b_j)], \quad (1)$$

where O_j is the fluorescent protein output of the bactoneuron, X_i represents the input inducer concentration, w_{ji} represents the weight of the corresponding input X_i, and b_j represents the bias for the bactoneuron j. Although it is not clear how bias is connected in molecular terms in the bactoneuron, it was found that the bias is linearly related to the gene expression leakage from the promoter[36]. We designed the molecular genetic circuits of all the bactoneurons based on the signs of the 'weight' parameter in activation function equations for the corresponding bactoneuron (Extended Data Fig. 1). The positive and negative weight values in an activation function of a bactoneuron respectively correspond to the activation and repression of output reporter gene expression mediated by a specific inducer. Zero weight corresponds to an invariant relation.

The transcriptional genetic circuits (Extended Data Fig. 1) were built based on a set of synthetic promoters on which transcription factors LuxR, LacR, TetR and CI[41] bind either individually or simultaneously to hinder and/or activate transcription. The transcription factors and promoters and their interactions formed various feed-forward, feedback and combination mechanisms based on the bactoneuron circuit designs (Extended Data Fig. 1). The mode of transcription was controlled by the interactions of the transcription factors with appropriate input chemicals (AHL, IPTG and aTc) through allosteric regulation. The quantitative relationship between inputs and outputs

https://doi.org/10.1038/s41589-024-01711-4

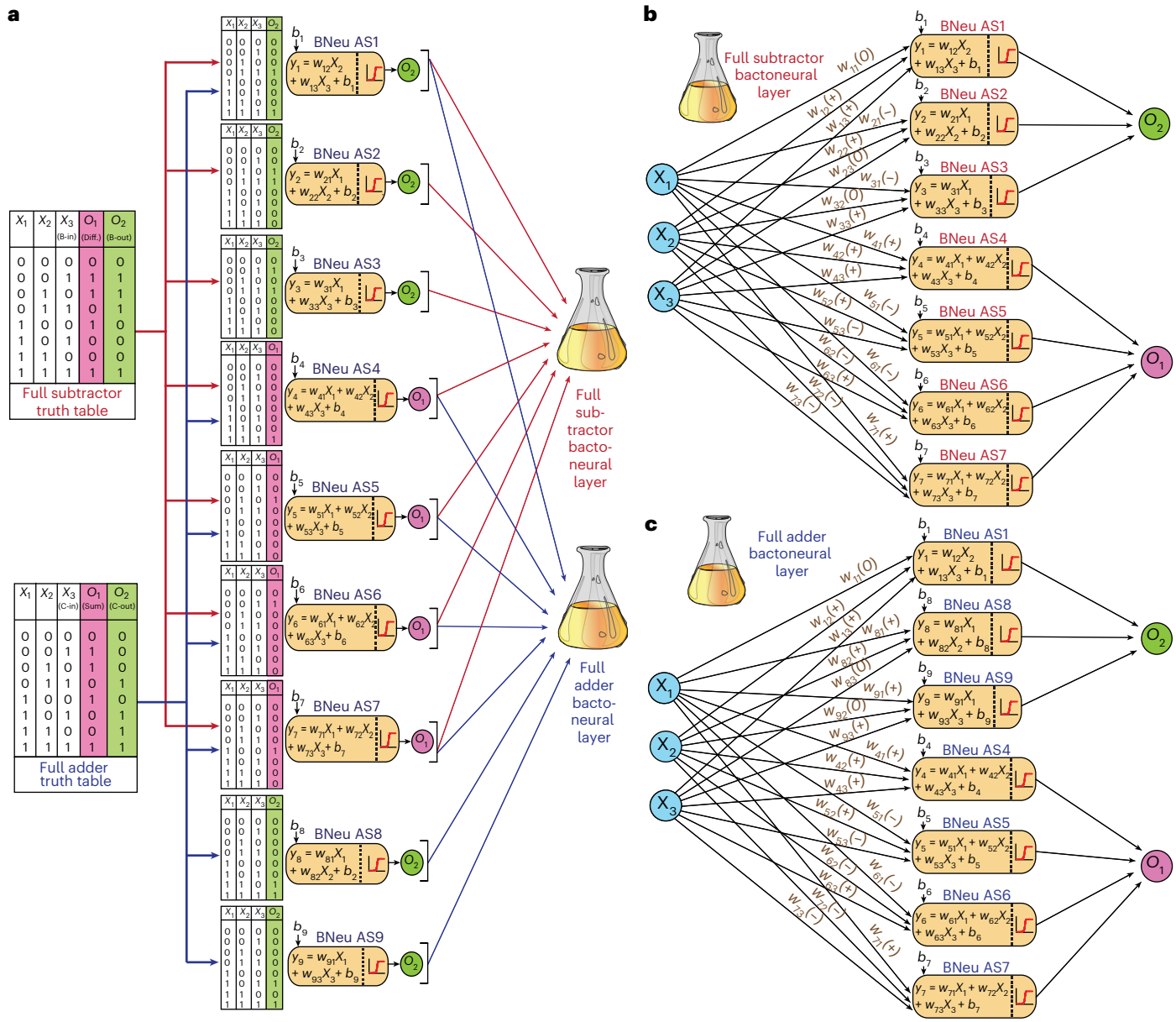


Fig. 1 | Mathematical and genetic network design. a, Deriving single bactoneurons from truth tables by fragmenting. The fragmented truth tables and corresponding bactoneurons generated out of a full subtractor and adder are indicated by red and blue connecting arrows, respectively. X_i and O_i represent the input and output, respectively, and w_{ij} represents the weight of the jth input for the ith bactoneuron. b_i represents the bias of the ith bactoneuron; B-in, borrow in; BNeu, bactoneuron; C-in, carry in; C-out, carry out; Diff., difference; B-out, borrow out. **b**,**c**, Configuration of the bactoneural single-layer network for the full subtractor (**b**) and adder (**c**).

within a proper parameter range should demonstrate appropriate bactoneuron behavior.

Construction, characterization and optimization

Among the nine bactoneurons required to configure a subtractor or adder, the basic forms for five two-input bactoneurons (AS1, AS2, AS3, AS8 and AS9) were available from a bactoneuron library that we previously made[36,37]. The rest of the four bactoneurons (AS4, AS5, AS6 and AS7) are three input systems that were built from scratch. We first constructed the initial set of molecular constructs according to the design (Extended Data Fig. 1) and inserted them into DH5αZ1 strains[42] of *Escherichia coli*. Next, we measured fluorescent protein expression at '0' and '1' (saturated) inducer concentrations (Extended Data Figs. 2–5). These experiments yielded fold change (FC) values between the highest output signal (value 1) and the highest leakage (where the leakage was defined as the basal level expression of the reporter fluorescent proteins under the input conditions where output expression should be 0), total leakage (Σ L) from all possible inducer (0,1) combinations and variation in the highest signals among various bactoneurons (SV). Further, we performed dose–response experiments as a function of various inducer concentrations (Extended Data Figs. 2–5) by varying one inducer concentration while keeping the others as constants (saturated or '0' concentration, as the case may be). The dose–response behaviors were fitted with the activation function equation (Eq. (1)), and the weight and bias parameters were estimated from the fitted equations (Extended Data Figs. 2–5). The initial constructs for AS5, AS6 and AS7 showed suboptimal behavior. AS5A, AS6A and AS7A generally had (1) low FC values between the highest output signal (value 1) and the highest leakage, (2) high Σ L, (3) low weight values and (4) high variability of the signals at ON states among various bactoneurons (SV; Fig. 2c).

We defined or fixed a set of optimum values arbitrarily or based on previous observations[37] (Fig. 2a). The minimum FC between the highest leakage and the ON state was set at 7. The maximum Σ L was set at 0.2 and 0.4 for two-input and three-input bactoneurons, respectively. The weight values define the sharpness of the transition from the OFF state to the ON state as a function of input. The minimum weight values were set as 8.5, 9.5 and 10.5 for inputs AHL, IPTG and aTc, respectively. Further, we measured the signal strength at ON states of all bactoneurons expressing a single fluorescent protein (EGFP or E2-Crimson) and fixed the maximum variation to be within 20%. Although bactoneuron AS4 satisfied the optimal set points in its first construct, the initial constructs for AS5, AS6 and AS7 failed. The measurements performed on all the two input bactoneurons taken from the library (AS1, AS2, AS3, AS8 and AS9) showed optimum behavior.

We performed a detailed failure analysis on the disqualified bactoneuron constructs (Supplementary Table 1). Now, keeping the general molecular design framework intact (Extended Data Fig. 1), we optimized those suboptimal bactoneurons by modifying the genetic constructs, altering the interaction strength among the various transcription factors and promoters and modulating their relative amounts in the cell. This had been done by changing plasmid copy numbers, using mutated forms of transcription factors, altering ribosome binding sites (RBSs) and promoter sequences and redistributing the bioparts (Fig. 2b). Guided by the optimization flow chart (Fig. 2a) and the failure analysis, optimization was conducted in multiple iterations (Fig. 2b) until quantitative set points were reached (Fig. 2c). The schematics of genetic construct and biopart distributions among various plasmids in each iteration for each bactoneuron are shown in Extended Data Figs. 6–8.

Simulation and validation of optimized bactoneurons

Based on the fitted parameters for each optimized bactoneuron, we performed computational simulations using parameterized activation functions (Fig. 3). Further, we experimentally tested the behavior of each optimized bactoneuron by simultaneously changing the concentrations of combinations of two inputs from among AHL, IPTG and aTc and measuring the fluorescence at each data point. To reduce the experimental bias, in each case, a different set of inducer concentrations was selected compared to the dose–response experiments (Supplementary Tables 2 and 5). These validation tests showed a close topological match between simulated and experimental behavior (Fig. 3). However, the simulation and experiment for aTc/IPTG for AS7 in Fig. 3i do not agree very well. One probable reason is that, unlike other bactoneurons (Extended Data Figs. 7 and 8) where we used one weak CI (CI*) and one strong CI (CI^w), in AS7 (Extended Data Fig. 8), we used two copies of strong CI^w repressors. We reasoned that for the same inducer concentration, the repression from two CI^w is higher than that of the other case (CI* + CI^w). Therefore, even with the low inducer concentrations, the repression is prominent in AS7, resulting in an inclined top rather than a flat top.

Mix and match configurable adder and subtractor

The nine optimized bactoneurons now worked as a configurable cellular system that could generate a full subtractor or a full adder. By mixing bactoneurons AS1, AS2, AS3, AS4, AS5, AS6 and AS7 in a coculture, we created a bactoneural layer that harbored the full subtractor function. Next, the subtractor coculture was exposed to various combinations of input chemicals (AHL, IPTG and aTc) according to the truth table, and the output fluorescence was observed using a fluorescence microscope. For the full subtractor, aTc served as the Borrow input, AHL served as the minuend, and IPTG served as the subtrahend. E2-Crimson expression (output O_1) signified difference, and EGFP expression (O_2) signified borrow-out. The results showed proper subtractor behavior (Fig. 4a). Similarly, by just replacing AS2 and AS3 with AS8 and AS9, we could reconfigure the bactoneural layer for a full adder behavior, where aTc and IPTG served as the general inputs and AHL served as the carry input; E2-Crimson expression signified sum, and EGFP expression signified carry-out. The experimental results showed logical adherence to the truth table of adder function under a fluorescence microscope (Fig. 4b).

Bacterial ANNs answer yes/no decision problems

Here, we created multiple cellular systems using a set of bactoneurons from the adder and subtractor and adding a few more bactoneurons, which, in a liquid culture, processes environmental chemical signals and solves a set of decision computational problems[43,44]. First, we designed four bacterial ANNs, each of which can answer a single yes/no decision problem. The problems include identifying prime numbers, perfect power numbers between 0 and 9, connected Haar graph numbers between 1 and 9 and vowels from a string of letters between A and L. In a bipartite graph, the vertices can be divided into two disjointed sets. Now, a Haar graph H(n) can be obtained by a simple binary encoding of cyclically adjacent vertices indexed by a positive integer n. For some n values, the H(n) are connected, like H_3 and H_5 (Fig. 5e), and those n are called connected Haar graph numbers. A perfect power number is a natural number that is equal to n^k, such that $n > 1$ and $k > 1$, where both n and k are natural numbers; 0 and 1 are also considered perfect power numbers, as $0^k = 0$ and $1^k = 1$ for any value of k.

To frame those problems in a chemical space by adding or not adding certain chemical inducers, first, we converted the decimal numbers from 0 to 9 into binary forms (Fig. 5a,c,e). Those binary numbers can be physically realized in a liquid culture by allowing four input chemicals. For example, a random question may be asked ('if 7 is a prime number') by adding IPTG, aTc and AHL but not adding arabinose (ARA) in the culture, where AHL represents the 'least significant bit', aTc represents the second bit, IPTG represents the third bit, and ARA represents the 'most significant bit' (0111) in the truth table (Fig. 5a). Similarly, letters from A to L have been converted to corresponding decimal numbers based on their position in the alphabet, and those decimal numbers were converted to corresponding binary numbers (Fig. 5g). The bacterial ANN would process those chemical signals, and the 'yes or no' answers were 'printed' by expressing appropriate fluorescent proteins. For example, in a prime number checker, a yes answer for a prime number like 7 is read by the green fluorescent protein, and a no answer for a nonprime number like 8 will cause the bacteria to express E2-Crimson (Fig. 5a).

The ANN design of the prime number identifier (Fig. 5a) required a new bactoneuron AS10 and the replacement of the output protein in AS2 from E2-Crimson to EGFP (AS2*) in addition to the previous four bactoneurons AS6–AS9. Similarly, to solve the other three problems with the ANN design, we built and added three new bactoneurons, AS11, AS12 and AS13, to the library and changed the output gene *egfp* of AS11 to the *mtagbfp2* gene, which encodes a blue fluorescent protein, to yield AS11*.

We designed, built and characterized the new bactoneurons AS10, AS11, AS12 and AS13 (Extended Data Figs. 9 and 10). Mixing and matching these new bactoneurons with old bactoneurons in a culture according to the ANN designs created single-bactoneural layers that harbored the prime number identifier (Fig. 5a), perfect power number identifier (Fig. 5c), connected Haar graph number identifier (Fig. 5e) and if a given letter between A and L is a vowel (Fig. 5g). The systems were exposed to various combinations of input chemicals (AHL, IPTG, aTc and ARA) to ask those questions, and the yes/no answers were read with a fluorescence microscope to identify prime numbers (Fig. 5b), perfect power numbers (Fig. 5d), connected Haar graph numbers (Fig. 5f) and vowels (Fig. 5h).

Each bacterial ANN gives multiple yes/no decisions

Next, we constructed two bacterial ANNs (Fig. 6a,c) such that each system answers more than one mathematical question that satisfies specific mathematical relations. One such question is, when we add 3 to an integer (n), do we get a prime number? For example, for $n = 2$, $n + 3$

https://doi.org/10.1038/s41589-024-01711-4

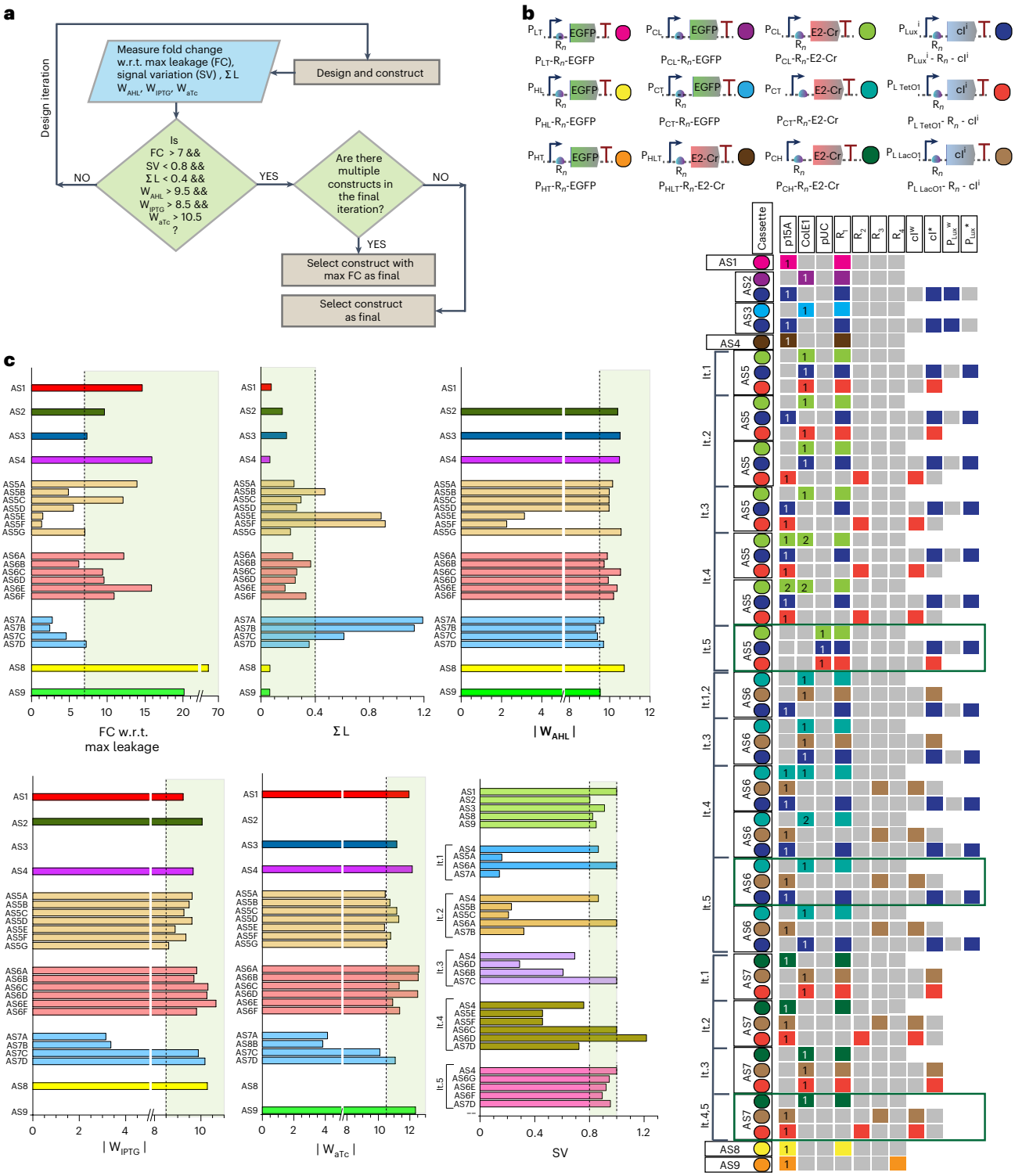


Fig. 2 | Optimization of bactoneuron parameters. a, Flow chart for bactoneural parameter optimization, where W_{AHL}, W_{IPTG} and W_{aTc} are the weight parameters for AHL, IPTG and aTc, respectively. The design and constructs are described in detail in Supplementary Fig. 1 and Extended Data Figs. 6–9; w.r.t., with reference to. **b**, Top, gene expression cassette configuration. Cassettes are indicated with color-coded circles. $P_{LT/HL/CL/CT/HLT/CH/Lux/L\ TetO1/L\ LacO1}$ denotes various promoters. Bottom, biological designs were iterated through modulating the relative copy number of the cassette-carrying plasmids (pUC/ColE1/p15A) and/or the cassette's RBS strength ($R_1/R_2/R_3/R_4$) and/or the repression strength of the λ repressor with wild-type cI^w/mutated cI^* and/or the promoter strength $P_{Lux}{}^w$ or $P_{Lux}{}^*$. The specific biopart configuration used for the gene expression cassette is indicated by squares that glow with the same color as the cassette color code. A gray square indicates an unused configuration. The number within each square indicates the number of that specific genetic cassette within a specific plasmid with a specific origin of replication (p15A, ColE1 and pUC) as mentioned at the top of each column. For bactoneurons that went through multiple design iterations, the corresponding constructs generated at each iteration are grouped under their iteration number, denoted by 'It.*n*', where *n* indicates the iteration number. Constructs that were chosen as final are highlighted in green boxes. Bactoneuron constructs under no iteration bracket denote no design iteration performed for them; E2-Cr, E2-Crimson. **c**, Parameter values achieved by different bactoneuron constructs. The set threshold and acceptable regions for each parameter are indicated by the dashed lines and the light green-shaded areas, respectively.

https://doi.org/10.1038/s41589-024-01711-4

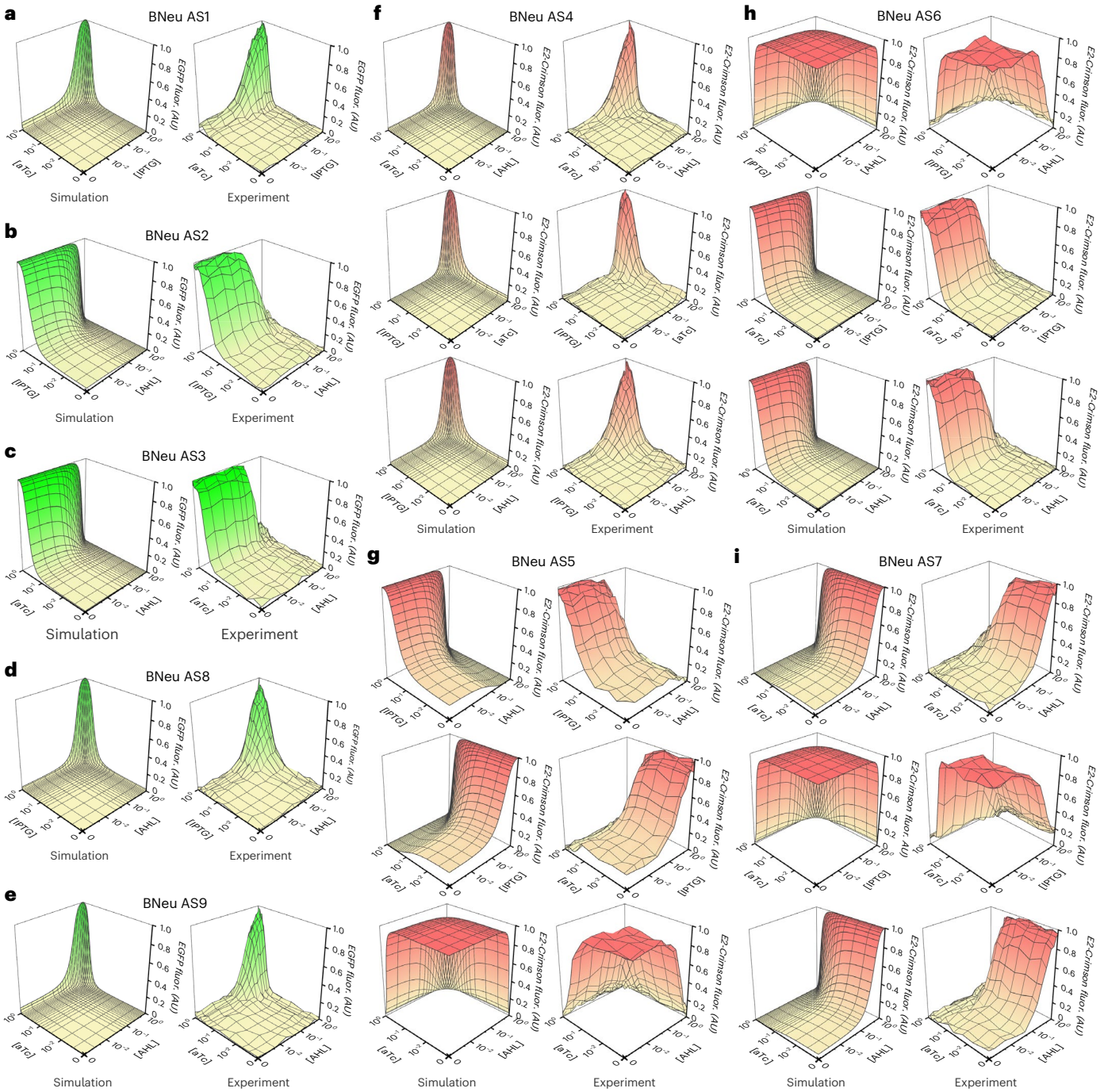


Fig. 3 | Simulations and experimental validation of optimized bactoneurons. **a**–**i**, Simulations and experimental validations for each bactoneuron are shown on the left and right, respectively. Each data point in the experimental validation represents a single data point. Data from bactoneurons AS1 (**a**), AS2 (**b**), AS3 (**c**), AS8 (**d**), AS9 (**e**), AS4 (**f**), AS5 (**g**), AS6 (**h**) and AS7 (**i**) are shown. Bactoneurons connected to an EGFP output channel are represented in a green-shaded topology, and those connected to E2-Crimson are shown in a red-shaded topology; AU, arbitrary units; fluor, fluorescence. Brackets indicate compound concentration.

(= 5) gives a prime number. Another such question is if the factorial of a number n is divisible by $n \times (n + 1)/2$. For example, 7! (= 5,040) is divisible by $7 \times (7 + 1)/2$ (= 28). Here, we designed and built a bacterial ANN (Fig. 6a) by mixing and matching bactoneurons AS5, AS10, AS12 and AS13, which can answer both questions simultaneously. The systems were exposed to various combinations of input chemicals (AHL, IPTG, aTc and ARA) to ask those two mathematical questions, and the answers were read with a fluorescence microscope such that if EGFP glowed, then the factorial of a number n is divisible by $n \times (n + 1)/2$, otherwise no. If E2-Crimson glowed, $n + 3$ is a prime number, otherwise no (Fig. 6b).

Similarly, we designed and built another bacterial ANN by mixing and matching bactoneurons AS2, AS5, AS11* (by changing the output EGFP of AS11 to mTagBFP2, a blue fluorescent protein) and AS13, resulting in a system (Fig. 6c) that answers three mathematical decision problems simultaneously. One question is if the square of a number n can be represented by the sum of three factorials. For example, $6^2 = 3! + 3! + 4!$. Another such question is if the square of a number is an interprime. Interprime is defined as the average of two successive prime numbers. For example, $n = 3$ satisfies the relation because 3^2 (= 9) is the average of two successive prime numbers 7 and 11. Similarly, the third question is

https://doi.org/10.1038/s41589-024-01711-4

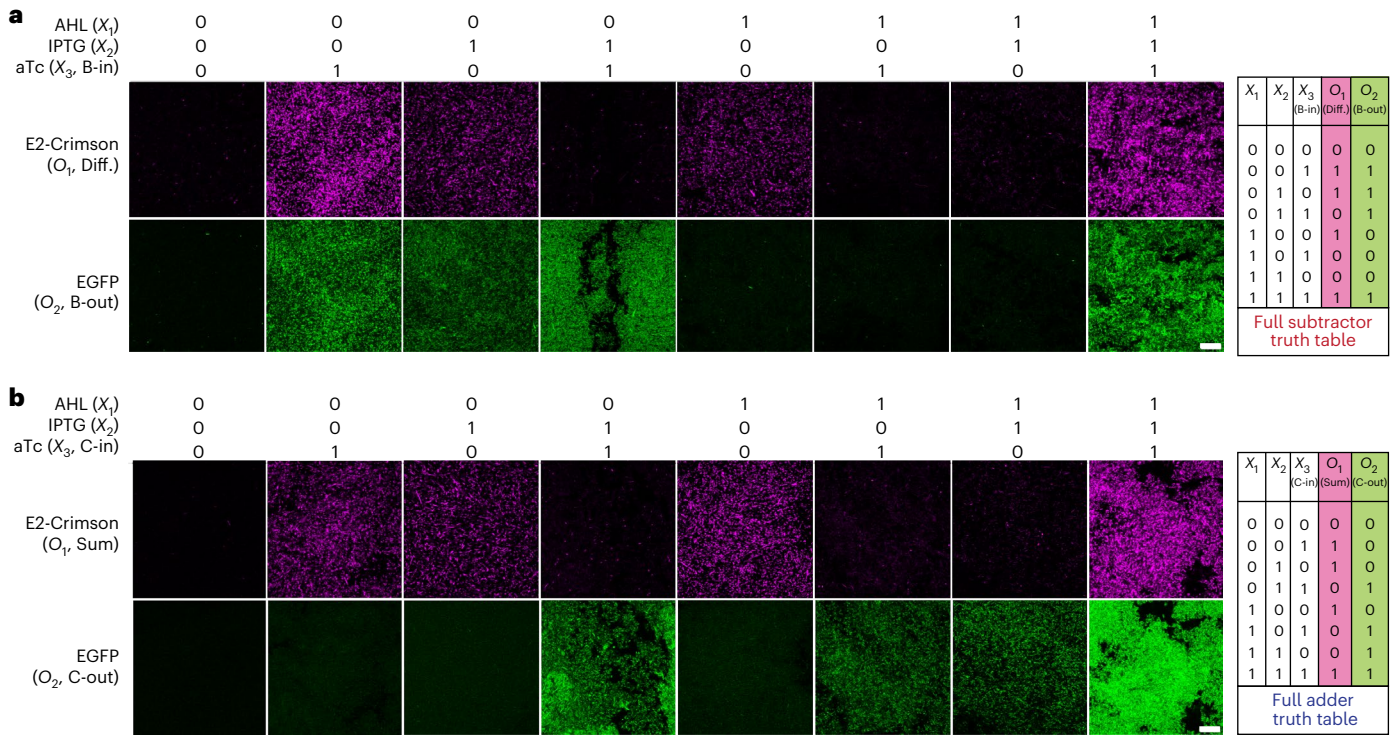


Fig. 4 | Fluorescence microscopy for full subtractor and full adder. Combinations of appropriate bactoneuron types as required to constitute the corresponding bactoneural layer were cocultured with appropriate inducer combinations (AHL, IPTG and aTc at 1 µM, 10 mM and 432 nM, respectively, as saturated concentrations for the 1 state and concentrations of 0 for the 0 state), as per the respective truth table (shown to the right). The cultures were washed with PBS and viewed as slides under relevant laser channels and emission filters (Methods); scale bar, 5 µm. Data are representative of four independent experiments and show the full subtractor (**a**) and full adder (**b**).

if n is a Ramanujan prime (Rn), which is the smallest number such that $\pi(x) - \pi(x/2) \geq n$ for all $x \geq$ Rn, where $\pi(x)$ is the prime counting function.

The systems were exposed to various combinations of input chemicals (AHL, IPTG, aTc and ARA) to ask those questions, and the yes/no answers were read with a fluorescence microscope (Fig. 6d) such that if mTagBFP2 glowed, the square of a number n can be represented by the sum of three factorials, otherwise no. If EGFP glowed, the square of that number (n) is an interprime, otherwise no, and if E2-Crimson glowed, the number n is a Ramanujan prime, otherwise no.

A bacterial ANN provides answers to a mathematical question

Another interesting problem is determining the maximum number of pieces from a pie (or any disc-shaped object) that can be obtained for a given number of straight cuts. This problem is different from yes/no decision problems and mimics an optimization problem[43,44]. The solution to this classic problem is known as the 'Lazy Caterer's Sequence' (The On-Line Encyclopedia of Integer Sequences; https://oeis.org/). The decimal number of straight cuts was communicated to the bacterial culture in binary form by adding or not adding three chemicals (AHL, IPTG and aTc; Fig. 6e). To satisfy the ANN design (Fig. 6e), we constructed a new bactoneuron AS14 (with EGFP) and one of its variants in output proteins mKO2 (an orange fluorescent protein AS14*; Extended Data Fig. 9), changed the output protein of AS1 from EGFP to E2-Crimson (AS1*) and mixed and matched them with AS1*, AS10, AS11* and AS12 (Fig. 6e), allowing us to demonstrate a function that determines the maximum number of pieces from a circular disc-type system like a pan cake or a pie for a given number of straight cuts. The answers could be read by visualizing the expression pattern of four fluorescent proteins, specifically E2-Crimson, EGFP, mTagBFP2 and mKO2 (Fig. 6f).

The fluorescence images in Figs. 4–6 showed different levels of fluorescence intensity in a few cases. This can be for a few reasons, including variations in ON-state cell concentrations due to variation in sample preparation or different growth rates in a coculture medium, variation in the number of cell types (bactoneurons) expressing fluorescent proteins in each condition and its ratio in the full ANN or variation in fluorescent protein expression among various cell types (bactoneurons).

Discussion

The idea of a cell-based biocomputer is in its infancy, and one of the key challenges is creating computational problem-solving capabilities[34]. We demonstrated a range of yes/no computational decision problem-solving tasks and one optimization problem by mixing and matching appropriate cell modules in a culture. To our knowledge, no engineered cellular system solved any of the computation problems demonstrated here and showed multiple computational problem-solving capacities.

Modularity and configurability help to manage complexity, perform parallel work and accommodate future uncertainty in engineering systems[45]. A good mix and match modular system should be able to demonstrate multiple functions by adding new modules, which can be used along with previous modules. In this work, each engineered cell, which represented an artificial neurosynapse within an ANN framework, worked as a modular and configurable system such that mixing and matching could generate a total of 12 functions, including 2 device functions, 6 cellular systems to answer nine computational decision problems and 1 system to answer a computational optimization problem.

Further, bactoneurons were integrated well and worked appropriately within multiple systems. For example, AS5 has been used in adder, subtractor, prime number checker, connected Haar graph number checker and two- and three-question answering systems.

https://doi.org/10.1038/s41589-024-01711-4

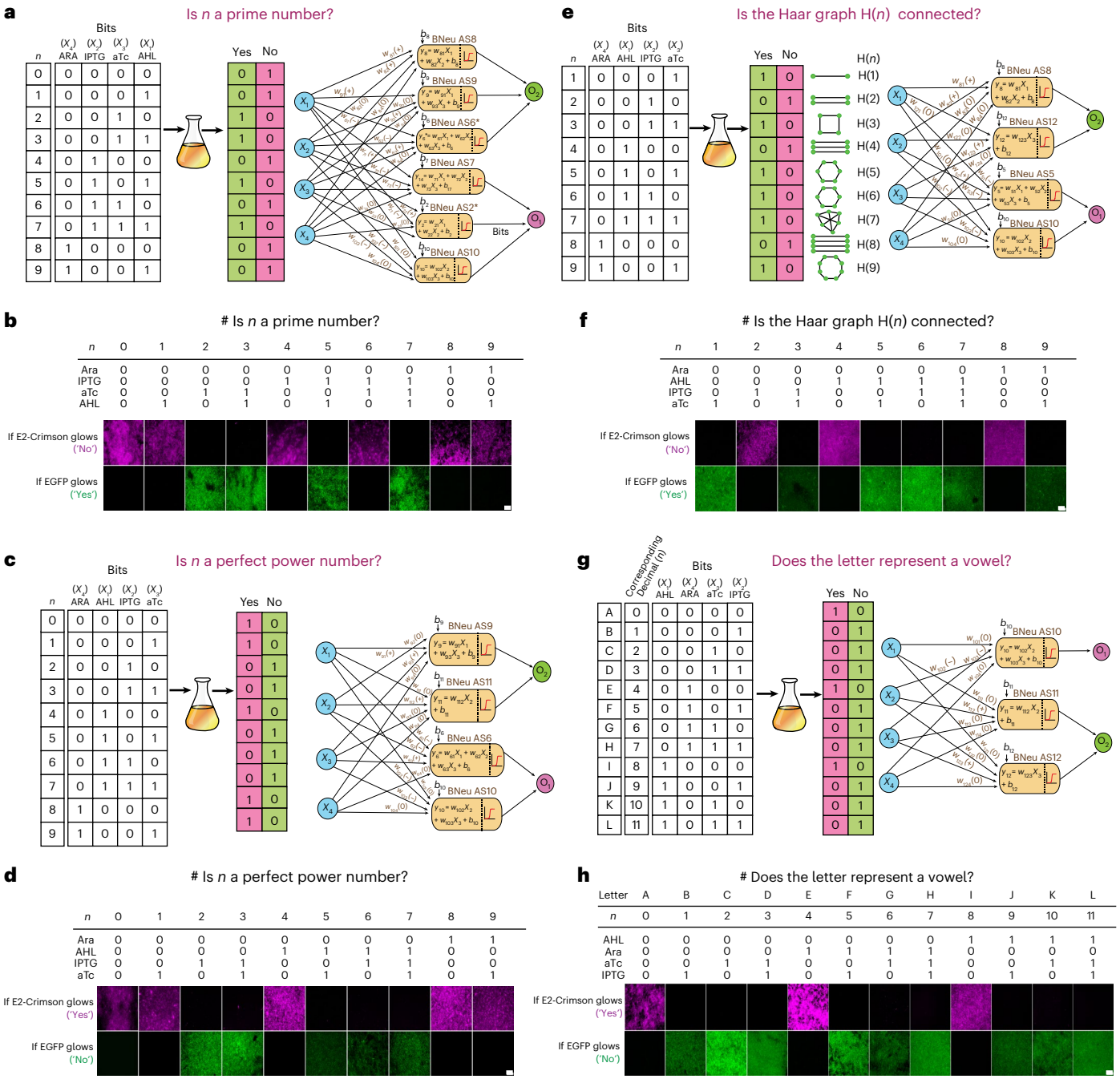


Fig. 5 | Bacterial ANNs answer yes/no computational problems. Representations of the problems in a biochemical space and bacterial ANN designs to answer the problems. **a**,**c**,**e**,**g**, If a number *n* between 0 and 9 is a prime number (**a**), if a number *n* between 0 and 9 is a perfect power number (**c**), if a Haar graph of number *n* between 1 and 9 H(*n*) is connected (**e**) and if a letter between A and L is a vowel (**g**). Four chemicals ARA, IPTG, aTc and AHL represent various bit positions in the problem matrix (truth table) for each problem. The chemical inputs (X_i) in bacterial ANNs are shown for each chemical for various problems on the top of the truth tables. Combinations of appropriate bactoneuron types as required to constitute the corresponding bactoneural layer were cocultured with appropriate inducer combinations (AHL, IPTG, aTc and ARA are at 1 µM, 10 mM, 432 nM and 6.66 mM, respectively, as saturated concentrations (1 state) and at concentrations of 0 (0 state)) to state the problems to the bacterial ANN as per the respective truth tables. **b**,**d**,**f**,**h**, Cultures were washed with PBS, and the answers were visualized under an epifluorescence microscope with relevant LED light excitation and emission filters (Methods) for prime numbers (**b**), perfect power numbers (**d**), connected Haar graph numbers (**f**) and vowels (**h**); scale bar, 50 µm. Data are representative of two independent experiments.

All bactoneurons except AS14 have been used in more than two cellular systems. This suggested that the modules were scalable in terms of integration with different functional systems. This system also allowed for scaling up the number of inputs and outputs for the new functions. We demonstrated the three-input–two-output systems (Fig. 4), four-input–two-output systems (Fig. 6a,b), four-input–three-output system (Fig. 6c,d) and three-input–four-output system (Fig. 6e,f).

The universal approximation theorem suggests that a feed-forward ANN-type architecture may approximate a wide variety of nonlinear functions[40]. Thus, an ANN architecture might be a good strategy to build complex biocomputing functions. In this work, all the computational functions were created through a single-layer ANN-type architecture, where no layering of the cells with chemical communication was required. Layering cells to obtain a function requires chemical

https://doi.org/10.1038/s41589-024-01711-4

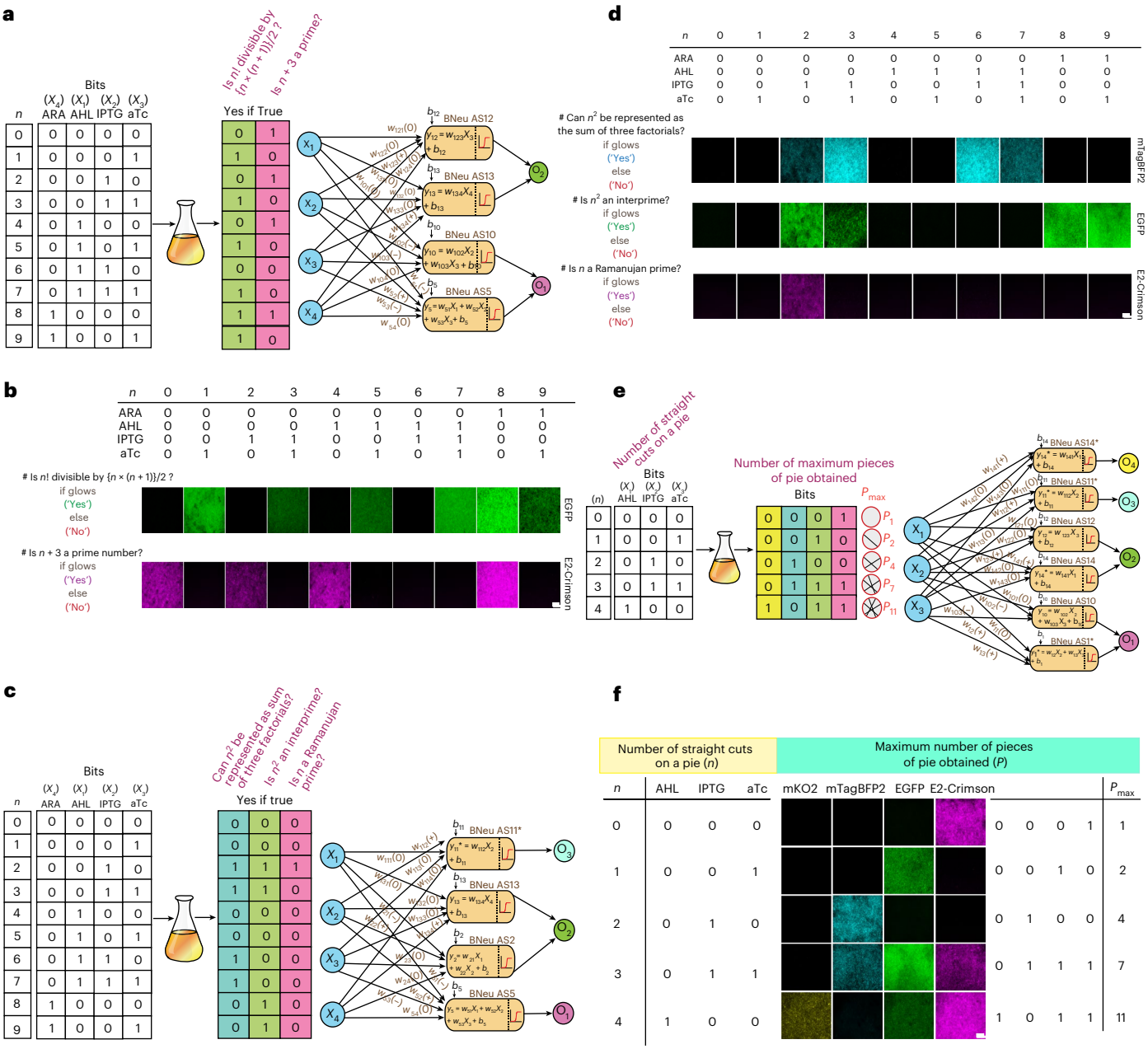


Fig. 6 | Bacterial ANNs answer multiple mathematical problems together and determine the maximum number of pieces obtained from a pie for a given number of straight cuts. a, The design and problem representation of a bacterial ANN answers two mathematical questions: if a factorial of a number n between 0 and 9 is divisible by $n \times (n+1)/2$ and if $n+3$ is a prime number. **b**, Answers (yes if it glows and no otherwise) to the two questions were visualized under an epifluorescence microscope (Methods). Representative microscopy results from two independent experiments are shown. **c**, Another bacterial ANN answers three mathematical questions together: if the square of a number n can be represented as factorial of three integers, if the square of a number is an interprime and if n is a Ramanujan prime. **d**, The answers (yes if it glows and no otherwise) to the three questions were visualized under an epifluorescence microscope (Methods). Representative microscopy results from two independent experiments are shown. **e**, Problem representation of 'what is the maximum number of pieces obtained from a pie for a given number of straight cuts?' and its bacterial ANN design. **f**, Answers were visualized under an epifluorescence microscope. The expression pattern of four fluorescent proteins gives the number in binary form, where E2-Crimson represents the 'least significant bit' and mKO2 represents the 'most significant bit'. Representative microscopy results from three independent experiments are shown; scale bar, 50 µm.

communication between layers in such a way that chemicals produced by one cell are the input for the next layer of cells. Proper tuning of those layered connections is a challenge[46]. Further, the number of commutation channels depends on the availability of the orthogonal diffusible chemical communication molecules and their corresponding regulatory genetic systems. Many times, the regulatory system consists of many genes[46], which are difficult to engineer in cells. Thus, single-layer designs are more advantageous than multilayer designs. Through this ANN approach, we demonstrated 12 computational functions by mixing and matching appropriate cells in a single layer.

However, it may also be possible to add more layers to this bacterial ANN. This can be achieved by replacing the output fluorescent protein genes with appropriate quorum-sensing genes, like *rhlI*/*lasI*/*cinI*, which produce diffusible quorum-sensing molecules[47]. Those molecules would diffuse and be taken by the second layer of bactoneurons to bind to responsive proteins like LasR/TraR/CinR and direct the native

or engineered promoters[46]. To build a much bigger circuit like a four-bit ripple carry adder or four-bit ripple borrow subtractor, combining chemical wires and optogenetics could be an effective strategy[48,49] but challenging.

Theoretical studies[50,51] predicted that to perform complex and large computations, multicellular distributed computing was necessary for its scalability, error tolerance and burden reduction per individual cell. One of the hurdles in building functional multicellular systems is the unavailability of quantitative parameters, which are required for understanding and controlling the system[52–54]. In this work, each of the modular bactoneurons was quantitatively characterized and mathematically predictable. As the whole ANN architecture related to its modules (nodes) through a general mathematical framework, the behavior of the ultimate device functions and computational problem solving were also predictable. Thus, the physical scaling up of the whole system was achieved by directly applying mathematical scaling up of the cellular modules.

Together, this work may have importance in biocomputer technology development, multicellular synthetic biology and cellular hardware for ANNs with potential industrial and therapeutic applications.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41589-024-01711-4.

References

1. Benenson, Y. Biomolecular computing systems: principles, progress and potential. *Nat. Rev. Genet.* **13**, 455–468 (2012).
2. Reif, J. H., Hauser, M., Pirrung, M. & LaBean, T. in *Complex Systems Science in Biomedicine* (eds. Deisboeck, T. S. & Kresh, J. Y.) 701–735 (Springer, 2006).
3. Ma, Q. et al. DNA computing: principle, construction, and applications in intelligent diagnostic. *Small Struct.* **2**, 2100051 (2021).
4. Yoon, J., Lim, J., Shin, M., Lee, T. & Choi, J. W. Toward bioelectronic device based on bionanohybrid composed of nanomaterials and biomaterials: from nucleic acid and protein to living cell. *Appl. Phys. Rev.* **10**, 011302 (2023).
5. Fan, D., Wang, J., Wang, E. & Dong, S. Propelling DNA computing with materials' power: recent advancements in innovative DNA logic computing systems and smart bio-applications. *Adv. Sci.* **7**, 2001766 (2020).
6. Lin, K. N., Volkel, K., Tuck, J. M. & Keung, A. J. Dynamic and scalable DNA-based information storage. *Nat. Commun.* **11**, 2981 (2020).
7. Lederman, H., Macdonald, J., Stefanovic, D. & Stojanovic, M. N. Deoxyribozyme-based three-input logic gates and construction of a molecular full adder. *Biochemistry* **45**, 1194–1199 (2006).
8. Fratto, B. E., Lewer, J. M. & Katz, E. An enzyme based half-adder and half-subtractor with a modular design. *ChemPhysChem* **17**, 2210–2217 (2016).
9. Cherry, K. M. & Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* **559**, 370–376 (2018).
10. Adleman, L. M. Molecular computation of solutions to combinatorial problems. *Science* **266**, 1021–1024 (1994).
11. Faulhammer, D., Cukras, A. R., Lipton, R. J. & Landweber, L. F. Molecular computation: RNA solutions to chess problems. *Proc. Natl Acad. Sci. USA* **97**, 1385–1389 (2000).
12. Chao, J. et al. Solving mazes with single-molecule DNA navigators. *Nat. Mater.* **18**, 273–279 (2018).
13. Brophy, J. A. N. & Voigt, C. A. Principles of genetic circuit design. *Nat. Methods* **11**, 508–520 (2014).
14. Bonnet, J., Yin, P., Ortiz, M. E., Subsoontorn, P. & Endy, D. Amplifying genetic logic gates. *Science* **340**, 599–603 (2013).
15. Bonnerjee, D., Mukhopadhyay, S. & Bagh, S. Design, fabrication, and device chemistry of a 3-input-3-output synthetic genetic combinatorial logic circuit with a 3-input AND gate in a single bacterial cell. *Bioconjug. Chem.* **30**, 3013–3020 (2019).
16. Can, U. I., Nagarajan, N., Vural, D. C. & Zorlutuna, P. Muscle-cell-based 'living diodes'. *Adv. Biosyst.* **1**, 1600035 (2017).
17. Wong, A., Wang, H., Poh, C. L. & Kitney, R. I. Layering genetic circuits to build a single cell, bacterial half adder. *BMC Biol.* **13**, 40 (2015).
18. Friedland, A. E. et al. Synthetic gene networks that count. *Science* **324**, 1199–1202 (2009).
19. Sexton, J. T. & Tabor, J. J. Multiplexing cell–cell communication. *Mol. Syst. Biol.* **16**, e9618 (2020).
20. Ausländer, D. et al. Programmable full-adder computations in communicating three-dimensional cell cultures. *Nat. Methods* **15**, 57–60 (2018).
21. Weinberg, B. et al. Large-scale design of robust genetic circuits with multiple inputs and outputs for mammalian cells. *Nat. Biotechnol.* **35**, 453–462 (2017).
22. Müller, M. et al. Designed cell consortia as fragrance-programmable analog-to-digital converters. *Nat. Chem. Biol.* **13**, 309–316 (2017).
23. Kim, M., Julius, A. A. & Cheang U. K. *Microbiorobotics: Biologically Inspired Microscale Robotic Systems* 2nd edn (Elsevier, 2017)
24. Justus, K. B. et al. A biosensing soft robot: autonomous parsing of chemical signals through integrated organic and inorganic interfaces. *Sci. Robot.* **4**, eaax0765 (2019).
25. Grozinger, L. et al. Pathways to cellular supremacy in biocomputing. *Nat. Commun.* **10**, 5250 (2019).
26. Chen, Y. Y. & Smolke, C. D. From DNA to targeted therapeutics: bringing synthetic biology to the clinic. *Sci. Transl. Med.* **3**, 106ps42 (2011).
27. Yan, X., Liu, X., Zhao, C. & Chen, G. Q. Applications of synthetic biology in medical and pharmaceutical fields. *Signal Transduct. Target. Ther.* **8**, 199 (2023).
28. Johnson, M. B., March, A. R. & Morsut, L. Engineering multicellular systems: using synthetic biology to control tissue self-organization. *Curr. Opin. Biomed. Eng.* **4**, 163–173 (2017).
29. Teravest, M. A., Li, Z. & Angenent, L. T. Bacteria-based biocomputing with cellular computing circuits to sense, decide, signal, and act. *Energy Environ. Sci.* **4**, 4907–4916 (2011).
30. Dvořák, P., Nikel, P. I., Damborský, J. & de Lorenzo, V. Bioremediation 3.0: engineering pollutant-removing bacteria in the times of systemic biology. *Biotechnol. Adv.* **35**, 845–866 (2017).
31. Gilbert, C. & Ellis, T. Biological engineered living materials: growing functional materials with genetically programmable properties. *ACS Synth. Biol.* **8**, 1–15 (2019).
32. Sarkar, K., Chakraborty, S., Bonnerjee, D. & Bagh, S. Distributed computing with engineered bacteria and its application in solving chemically generated 2 × 2 maze problems. *ACS Synth. Biol.* **10**, 2456–2464 (2021).
33. Collins, J. Synthetic biology: bits and pieces come to life. *Nature* **483**, S8–S10 (2012).
34. Ren, X. et al. Cardiac muscle cell-based coupled oscillator network for collective computing. *Adv. Intell. Syst.* **3**, 2000253 (2021).
35. Ji, J. et al. Large-scale cardiac muscle cell-based coupled oscillator network for vertex coloring problem. *Adv. Intell. Syst.* **5**, 2200356 (2023).
36. Sarkar, K., Bonnerjee, D., Srivastava, R. & Bagh, S. A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing. *Chem. Sci.* **12**, 15821–15832 (2021).

37. Srivastava, R. & Bagh, S. A logically reversible double Feynman gate with molecular engineered bacteria arranged in an artificial neural network-type architecture. *ACS Synth. Biol.* **12**, 51–60 (2023).
38. Rizik, L., Danial, L., Habib, M., Weiss, R. & Daniel, R. Synthetic neuromorphic computing in living cells. *Nat. Commun.* **13**, 5602 (2022).
39. Millacura, F. A., Largey, B. & French, C. E. ParAlleL: a novel population-based approach to biological logic gates. *Front. Bioeng. Biotechnol.* **7**, 46 (2019).
40. Demuth, H. & De Jesús, B. *Neural Network Design* 2nd edn (Martin Hagan, 2014).
41. Sarkar, K., Mukhopadhyay, S., Bonnerjee, D., Srivastava, R. & Bagh, S. A frame-shifted gene, which rescued its function by non-natural start codons and its application in constructing synthetic gene circuits. *J. Biol. Eng.* **13**, 20 (2019).
42. Lutz, R. & Bujard, H. Independent and tight regulation of transcriptional units in *Escherichia coli* via the LacR/O, the TetR/O and AraC/I 1-I 2 regulatory elements. *Nucleic Acids Res.* **25**, 1203–1210 (1997).
43. Goldreich, O. *Computational Complexity: A Conceptual Perspective* (Cambridge Univ. Press, 2008).
44. Mezard, M. & Montanari, A. *Information, Physics and Computation* (Oxford Graduate Texts, Oxford Univ. Press, 2009).
45. Baldwin, C. Y. & Clark, K. B. in *Complex Engineered Systems* (eds. Braha, D. et al.) 175–205 (Springer, 2006).
46. Glykofrydis, F. & Elfick, A. Exploring standards for multicellular mammalian synthetic biology. *Trends Biotechnol.* **40**, 1299–1312 (2022).
47. Boo, A., Amaro, R. L. & Stan, G. B. Quorum sensing in synthetic biology: a review. *Curr. Opin. Syst. Biol.* **28**, 100378 (2021).
48. Ohlendorf, R., Vidavski, R. R., Eldar, A., Moffat, K. & Möglich, A. From dusk till dawn: one-plasmid systems for light-regulated gene expression. *J. Mol. Biol.* **416**, 534–542 (2012).
49. Multamäki, E. et al. Optogenetic control of bacterial expression by red light. *ACS Synth. Biol.* **11**, 3354–3367 (2022).
50. Macia, J., Vidiella, B. & Solé, R. V. Synthetic associative learning in engineered multicellular consortia. *J. R. Soc. Interface* **14**, 20170158 (2017).
51. Moškon, M., Pušnik, Ž., Zimic, N. & Mraz, M. Field-programmable biological circuits and configurable (bio)logic blocks for distributed biological computing. *Comput. Biol. Med.* **128**, 104109 (2021).
52. Beal, J. et al. The long journey towards standards for engineering biosystems. *EMBO Rep.* **21**, e50521 (2020).
53. Toda, S., Blauch, L. R., Tang, S. K. Y., Morsut, L. & Lim, W. A. Programming self-organizing multicellular structures with synthetic cell–cell signaling. *Science* **361**, 156–162 (2018).
54. Aydin, O. et al. Principles for the design of multicellular engineered living systems. *APL Bioeng.* **6**, 010903 (2022).

Methods

Ribosome binding site design and promoters, genes and plasmids

Promoters, RBSs, genes and transcription terminators were all assembled using the Network Brick pipeline[55] into pZ expression systems[42] according to the corresponding genetic circuit designs. RBSs with different strengths were designed, and translation initiation rates for designed RBSs (Supplementary Table 3) were calculated thermodynamically from an RBS calculator[56]. Standard molecular biology protocols were used. Restriction enzymes, T4 DNA ligase, buffers and DNA ladders were from New England BioLabs. PCR reactions were performed using the KOD Hotstart DNA polymerase (Merck Millipore). Plasmid isolation, PCR purification and gel extraction were performed using Qiagen kits. All promoter sequences, gene sequences and primer sequences are shown in Supplementary Table 4. Details of the cellular devices and plasmids are shown in Extended Data Figs. 6–9. Primers were synthesized by Integrated DNA Technologies. Oligonucleotides and gene products were synthesized by Invitrogen GeneArt Gene Synthesis service (Thermo Fisher). All constructs and bioparts were sequence verified by Eurofins Genomics.

Bacterial strains and culture conditions

For all cloning purposes, the *E. coli* DH5α strain was used. For all experiments, the DH5αZ1 strain[42] was used as the chassis organism. Chemically competent DH5αZ1 cells were cotransformed with sequence-verified plasmid constructs as per the design of each of the bactoneurons and were plated onto LB agar (Miller; Himedia) plates with appropriate antibiotics (ampicillin (Himedia) at 100 μg ml^{-1} and chloramphenicol (Himedia) at 34 μg ml^{-1}). For all liquid cultures, LB broth (Miller; Himedia) was used with appropriate antibiotics. Appropriate chemical inducers were provided into liquid cultures (AHL ($C_{10}H_{15}NO_4$; Merck), 0–5,000 nM; IPTG (Himedia), 0–10,000 μM; aTc (Sigma) 0–200 ng ml^{-1}). Incubation was performed at 37 °C and ~280 rpm in an Innova 42 incubator shaker (New Brunswick Scientific).

Testing of bactoneuron constructs for adherence to respective truth tables

Four distinct single colonies were picked from the plates and grown in liquid culture. For induction, overnight cultures were diluted (100 times) in fresh LB with appropriate inducer combinations as per the truth table (Fig. 1). This induction was given for a total of 16 h with a subculture of a 1:100 dilution at the 10-h time point. The 0 and 1 states signified zero and saturated inducer concentrations, respectively.

Dose–response experiments

Three or four single-colony overnight cultures for each bactoneuron were set up in LB broth as described before. For an experimental set, each of these cultures was subjected to a 16-h induction (with a subculture of a 1:100 dilution into identical composition medium at the 10-h time point) in separate 3-ml LB broth cultures with different inducer concentration points, where one inducer was varied across ten or more intermediate concentration points (Supplementary Table 2) between its 0 and 1 states (see scaling below), whereas the other inducer(s) were kept constant at 0 for a repressor or at saturated concentrations for an activator. This was performed for each bactoneuron for each of the three or two inducers with non-zero weights.

Fluorescence measurements, normalization and scaling

After the 16-h induction, cells were washed and resuspended in PBS (pH 7.4) for fluorescence measurements. Cells were resuspended in PBS such that optical density at 600 nm (OD_{600}) values were between 0.7 and 0.9. These cultures were then loaded onto a black, flat-bottom 96-well plate (Greiner), and, using a Synergy HTX Multi-Mode reader (Biotek Instruments), the OD_{600} and fluorescence were measured using appropriate combinations of excitation and emission filters. Cells were excited by a white light source that had been passed through an excitation filter of 485/20-nm bandpass for EGFP and 610/10-nm bandpass for E2-Crimson, while emission was collected by 516/20-nm bandpass filter for EGFP at gain 60 and 645/10-nm bandpass filter at gain 70 for E2-Crimson. To collect the fluorescence and OD data, at least three biological replicates were considered for each condition. To normalize the raw fluorescence reading obtained, for the number of cells in the well, the raw fluorescence values were divided by the respective OD_{600} values of each well. Autofluorescence was measured as the average normalized fluorescence of the untransformed DH5αZ1 set (no plasmid set) and was subtracted from the normalized fluorescence value of the experimental set. The resultant value was averaged across biological replicates to obtain the fluorescence intensity at an induction condition. The above process can be mathematically represented as

$$\begin{aligned}&\text{Unscaled normalized fluorescence}\\&=\left[\frac{(\text{Absolute flourescence value from experimental cell population})}{(\text{OD of experimental cell population})}\right]\\&-\left[\frac{(\text{Absolute flourescence value from no plasmid cell population})}{(\text{OD of no plasmid cell population})}\right]\end{aligned} \tag{2}$$

These normalized fluorescence values were then scaled down between 0 and 1 by considering the normalized fluorescence value at the induction point of maximum expected/observed fluorescence to be 1 and by dividing all other fluorescence values by this fluorescence value.

$$\begin{aligned}&\text{Scaled normalized fluorescence}\\&=\left[\frac{\begin{array}{c}(\text{Unscaled normalized fluorescence}\\ \text{at any induction point}\end{array}}{\begin{array}{c}(\text{Unscaled normalized fluorescence at induction point}\\ \text{of maximum expected and observed fluorescence})\end{array}}\right]\end{aligned} \tag{3}$$

Fitting and scaling input concentration points and simulation

The scaled fluorescence values (y axis) obtained were plotted against varying inducer concentration points (x axis) onto a scatter plot on Origin2018 (OriginLab). Using the in-built orthogonal distance regression algorithm of the software, the scatter plot was fitted with a modified form of Eq. (1),

$$O_j = \frac{1}{1+e^{-(w_j n.X_j n+B)}}, \tag{4a}$$

where O_j represents the normalized and scaled fluorescent outputs, $w_j n$ represents the weight of the nth inducer input for the jth bactoneuron, and $X_j n$ represents the scaled concentration for the nth input, and

$$B = w_{jm}.x_{jm} + w_{jp}.x_{jp} + b_j, \tag{4b}$$

where $w_{j(m,p)}$ and $x_{j(m,p)}$ are weights and inputs for the other two (m and p) inducers. B is a constant for an inducer-specific dose–response experiment for a specific bactoneuron. Inducer concentration points were scaled between 0 and 1, where 0 represents the absence of the inducer in the experimental set and 1 represents the concentration of the inducer where O_j attains a value of 0.999 for inducers with a positive weight sign or 0.001 for those with a negative weight sign. All values of concentration above this point were designated 1. The bias b_j value of the bactoneuron was derived by solving different parameterized versions of Eq. (4b) from different dose–response experiments. We have now chosen 44 points in between 0 and 1 to perform the simulation for each of the bactoneurons in this study (Supplementary Table 5).

For simulation, all bactoneurons were modeled based on the activation function (Eq. (1)). Parameterized activation functions with numerical values of w_j and b_j were used for computational simulations for each bactoneuron by generating 50 × 50 matrices of estimated output fluorescence values 'O_j', where two inputs simultaneously varied across 50 × 50 scaled relative concentration points between 0 and 1. The third inducer (if any) was kept at 0 or 1 as the case may be. We used a fixed set of scaled relative concentration points to draw all the simulations as listed in Supplementary Table 5.

Validation experiments for verifying mathematical models and simulations

Verification of the mathematical models and simulations was done through validatory experiments subjecting an overnight culture of a single colony ($n = 1$) of an appropriately transformed DH5αZ1 cell to a 16-h induction, as described previously, where two inducers were varied simultaneously across 16 × 16 concentration points (keeping the third inducer, if any, constant at 0 or 1 as the case may be) that were different from those that were used in the dose–response experiments. The list of concentration points for validation experiments for different inducers and bactoneurons is provided in Supplementary Table 2. Fluorescence readings were recorded, normalized, scaled and plotted against the scaled concentration points of two inducers.

Fluorescence microscopy

After overnight culture of each colony, we diluted (100×) each in fresh medium, and after culturing for 6 h, we measured the OD for each bactoneuron. During coculture, based on the bactoneural configuration, the dilution (~100×) with the appropriate inducer combinations as per the truth table for each bactoneuron was made in such a way that the starting concentration of each bactoneuron in the consortia was the same. We did not perform any experiments to compare the results of varying the concentration proportion of each bactoneuron in the starting consortium culture. The coculture was grown for 10 h. These 10-h cocultures were again subcultured with a 100-fold dilution for another 6 h, and the final cocultures were washed twice with PBS (pH 7.4). The final washed cell pellets were again resuspended in fresh PBS, added onto glass slides, covered with coverslips and sealed with a sealer. These were observed under either a confocal or fluorescence microscope. Specifically, cells were observed under a ×60 water immersion objective on a Nikon A1R Si confocal microscope supplied with a resonant scanner and coherent CUBE diode laser system. The sample fields were subjected to excitation of the fluorescent proteins by sequentially illuminating with two laser channels, with $\lambda = 488$ nm for EGFP and $\lambda = 561$ nm for E2-Crimson. Fluorescence emissions were captured through a T-PMT with relevant emission filters of 525/50-nm bandpass for EGFP and 700/75-nm bandpass for E2-Crimson. For epifluorescence, cells were observed under a 40× air objective under an EVOS cell imaging system M 5000 with four light cubes: (excitation/emission) 357/447 nm for mTagBFP2, 470/525 nm for EGFP, 531/593 nm for mKO2 and 585/624 nm for E2-Crimson. Differential interference contrast images were also acquired for each captured field. ImageJ Fiji software was used for the processing of acquired images.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All relevant data supporting the findings are available within the paper, Extended Data figures and the Supplementary Data. Source data are provided with this paper.

References

55. Mukhopadhyay, S., Sarkar, K., Srivastava, R., Pal, A. & Bagh, S. Processing two environmental chemical signals with a synthetic genetic IMPLY gate, a 2-input-2-output integrated logic circuit, and a process pipeline to optimize its systems chemistry in *Escherichia coli*. *Biotechnol. Bioeng.* **117**, 1502–1512 (2020).
56. Salis, H. M., Mirsky, E. A. & Voigt, C. A. Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.* **27**, 946–950 (2009).

Acknowledgements

This work was financially supported by grant RSI4002, Department of Atomic Energy, Government of India, awarded to Saha Institute of Nuclear Physics, and S.B. is a part of it.

Author contributions

D.B., S.C., B.M., R.B. and A.P. performed all the experiments. D.B., S.C., B.M., R.B. and S.B. analyzed the data. D.B., S.C. and S.B. designed the experiments and wrote the manuscript. S.B. designed and conceived the study.

Competing interests

The authors declare no competing interests.

Additional information

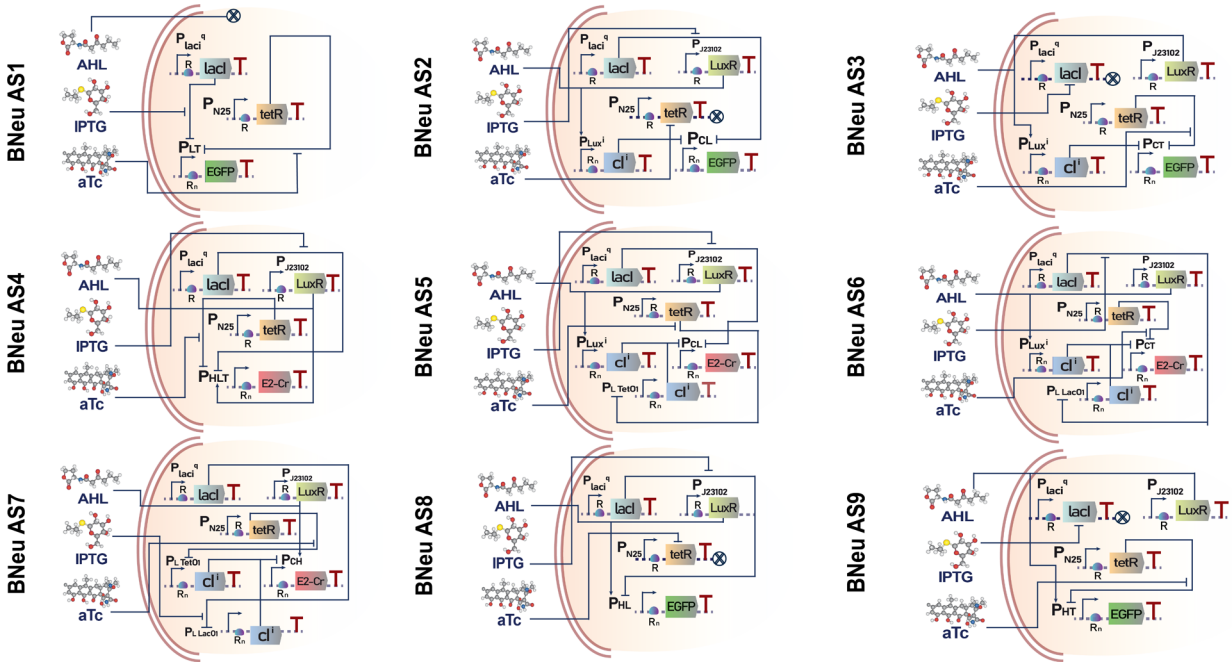
Extended data is available for this paper at https://doi.org/10.1038/s41589-024-01711-4.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41589-024-01711-4.

Correspondence and requests for materials should be addressed to Sangram Bagh.

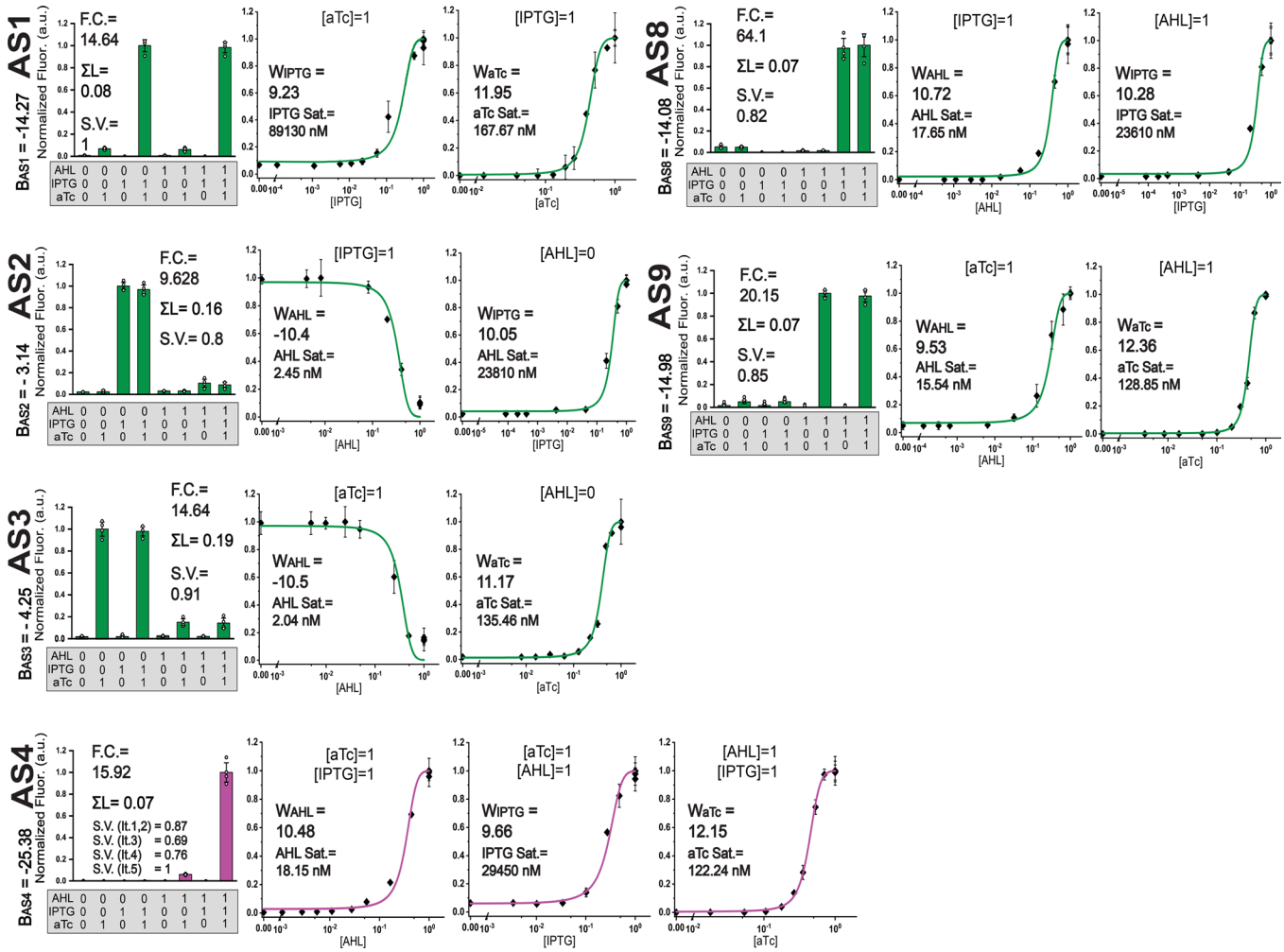
Peer review information *Nature Chemical Biology* thanks Allen Liu, Pinar Zorlutuna and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s41589-024-01711-4



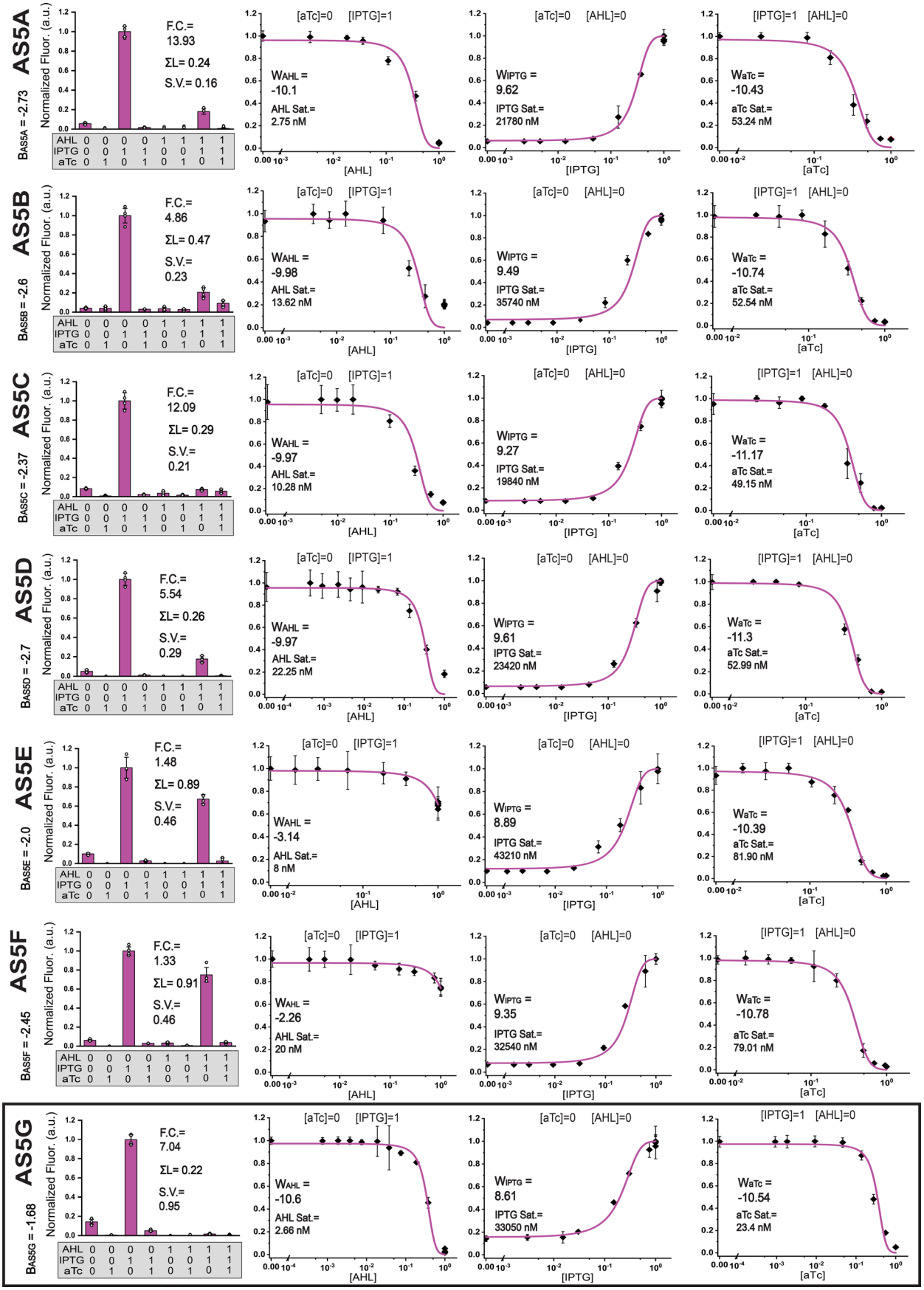
Extended Data Fig. 1 | Genetic circuit designs. Generic maps of the synthetic genetic networks that encode corresponding specific functions to each bactoneuron type.

https://doi.org/10.1038/s41589-024-01711-4



Extended Data Fig. 2 | Details of characterization and Dose Responses experiments of Bactoneurons that went through no iteration and were selected as final. The bactoneurons AS1, AS2, AS3, AS4, AS8 and AS9 were separately subjected to all possible combinations of input conditions (saturated concentration '1' or absent '0') yielding corresponding parameter values of Fold Change (F.C.), Σ Leakage (ΣL) and Signal variation (S.V.) (shown in figure). 'Signal variation' (S.V.) is a comparative parameter drawn by scaling the fluorescence signal (@ 'ON' state) of all bactoneurons in an iteration set (with the same output channel) by considering the strongest signal of all bactoneurons as '1'. For Dose response, the concentration of one parameter was varied across various concentration points while the other two inducers were kept constant at '0' for repressor or saturated concentration for activator (specified atop each dose response curve). The fluorescence readings obtained from the dose response experiments were fitted with the Log-Sigmoid function (see Methods) to derive the parameters Bias ($B_{AS\#}$) and Weights ($W_{inducer}$) and the respective Inducer Saturation points (Sat.) were identified. Each data point represent mean and s.d. from four independent colonies.

https://doi.org/10.1038/s41589-024-01711-4

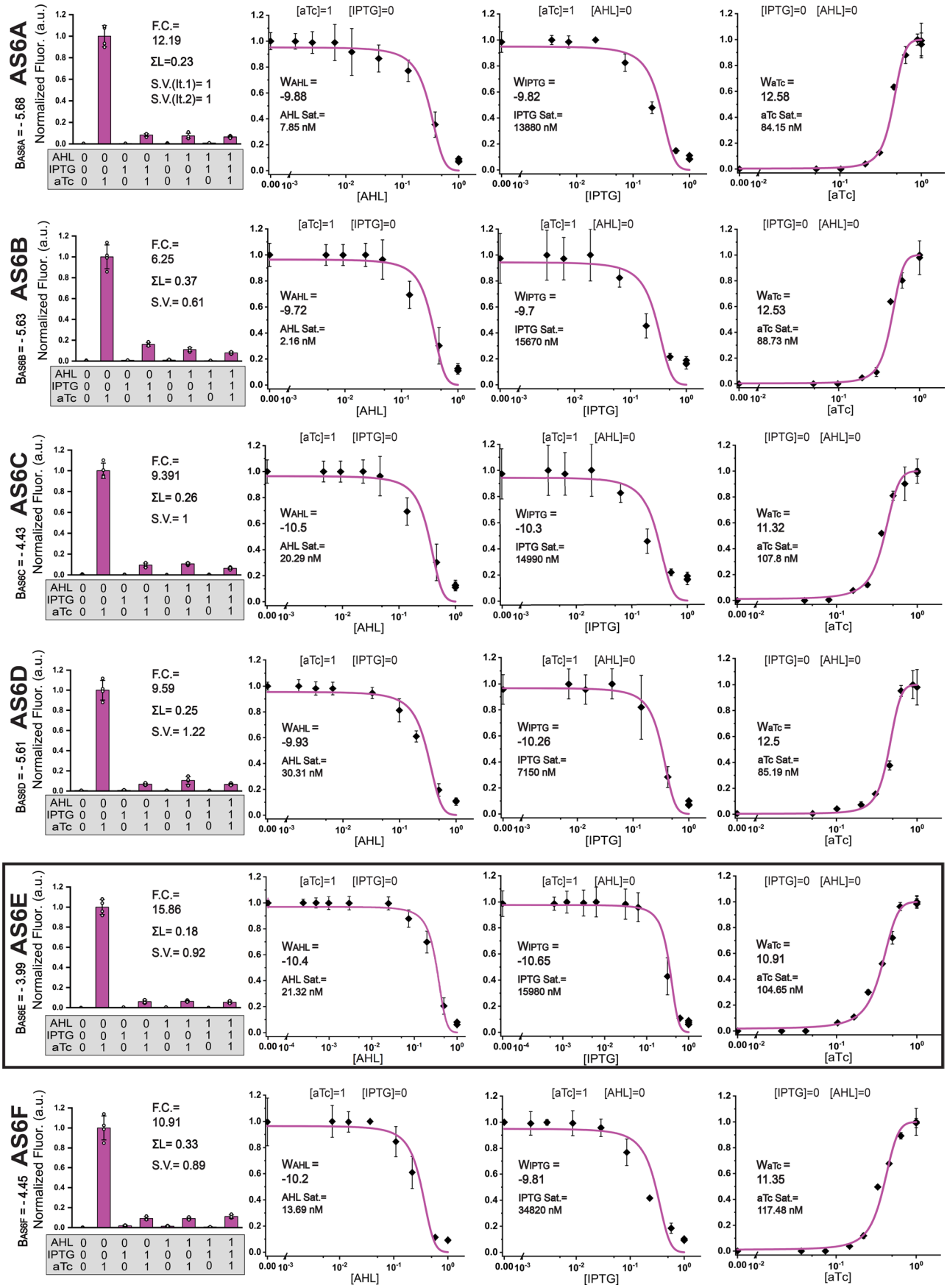


Extended Data Fig. 3 | See next page for caption.

https://doi.org/10.1038/s41589-024-01711-4

Extended Data Fig. 3 | Details of characterization and Dose Responses experiments across iterations of Bactoneuron 5. AS5A, AS5B, AS5C, AS5D, AS5E, AS5F and AS5G; yielding corresponding parameter values (shown in figure) of Fold Change (F.C.), Σ Leakage (ΣL), Signal variation (S.V.), Bias ($B_{AS\#}$), Weights ($W_{inducer}$) and Inducer Saturation points (Sat.). The fluorescence readings obtained from the dose response experiments were fitted with the Log-Sigmoid function (see Methods) to derive the Weight and Bias parameters. The induction states of the two constant inducers are specified atop the fitted curves. The final selected Construct is shown boxed. Each data point represent mean and s.d. from four independent colonies.

https://doi.org/10.1038/s41589-024-01711-4

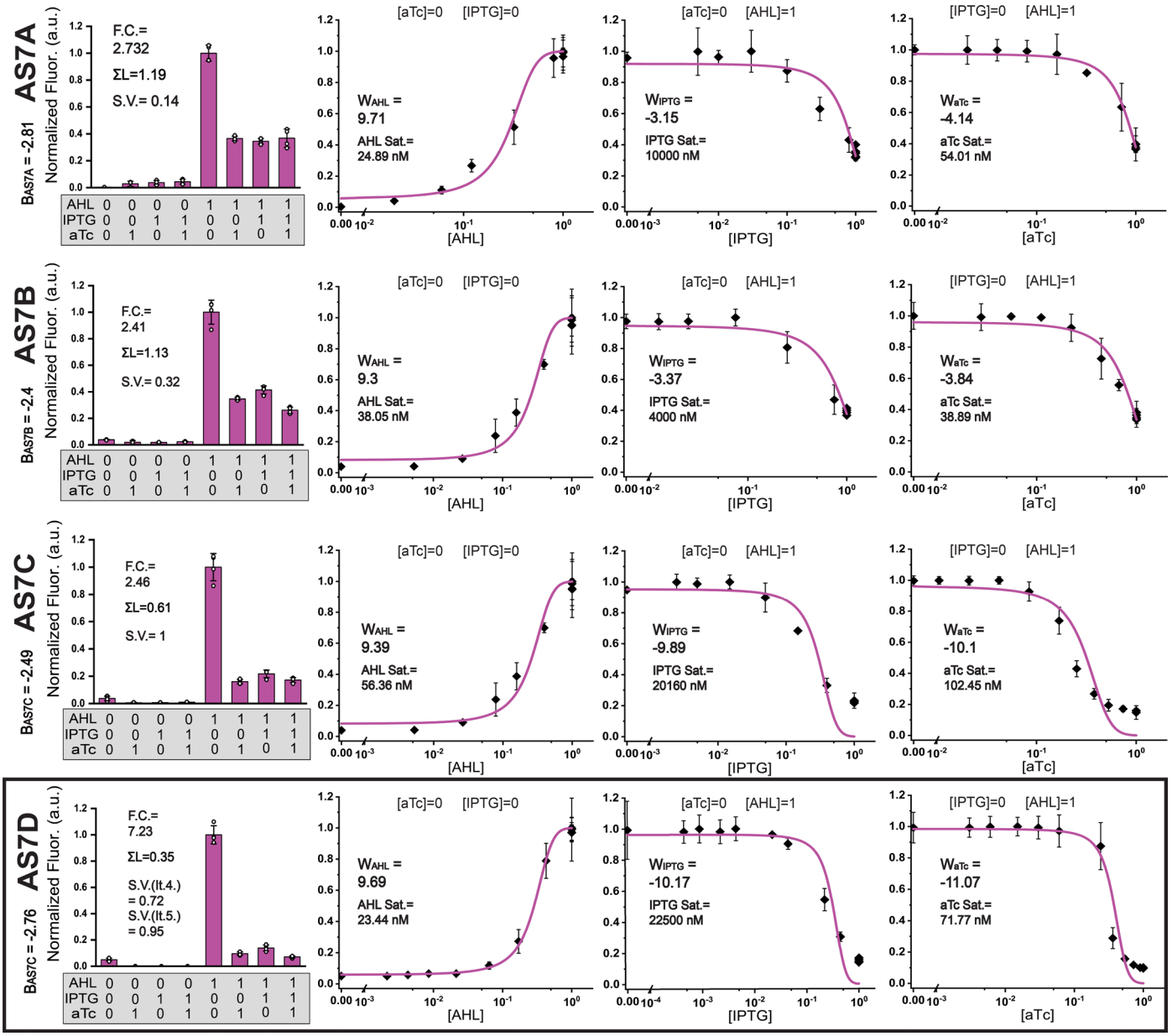


Extended Data Fig. 4 | See next page for caption.

https://doi.org/10.1038/s41589-024-01711-4

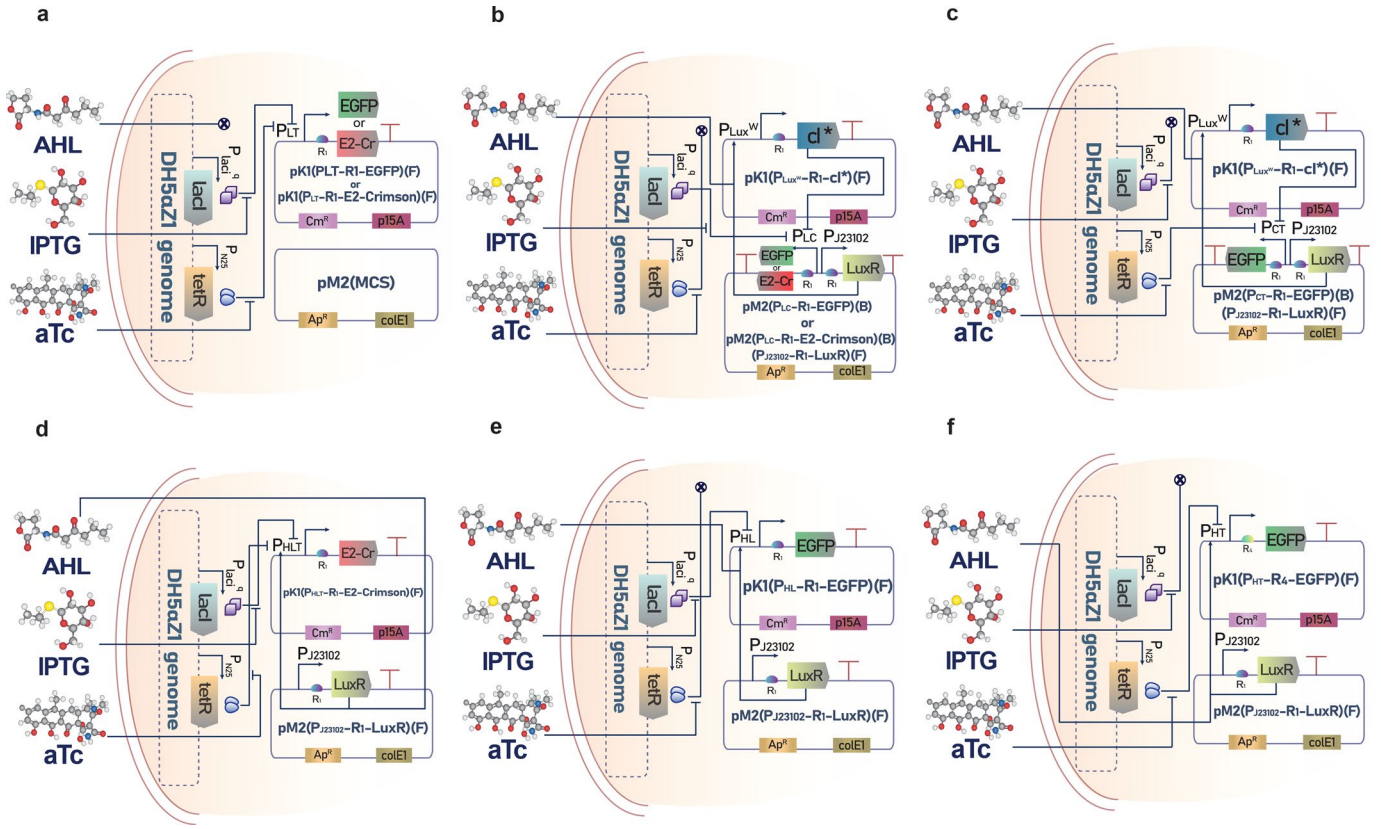
Extended Data Fig. 4 | Details of characterization and Dose Responses experiments across iterations of Bactoneuron 6. AS6A, AS6B, AS6C, AS6D, AS6E and AS6F; yielding corresponding parameter values (shown in figure) of Fold Change (F.C.), Σ Leakage (ΣL), Signal variation (S.V.), Bias ($B_{AS\#}$), Weights ($W_{Inducer}$) and Inducer Saturation points (Sat.). The fluorescence readings obtained from the dose response experiments were fitted with the Log-Sigmoid function (see Methods) to derive the Weight and Bias parameters. The induction states of the two constant inducers are specified atop the fitted curves. The final selected Construct is shown boxed. Each data point represent mean and s.d. from four independent colonies.

https://doi.org/10.1038/s41589-024-01711-4



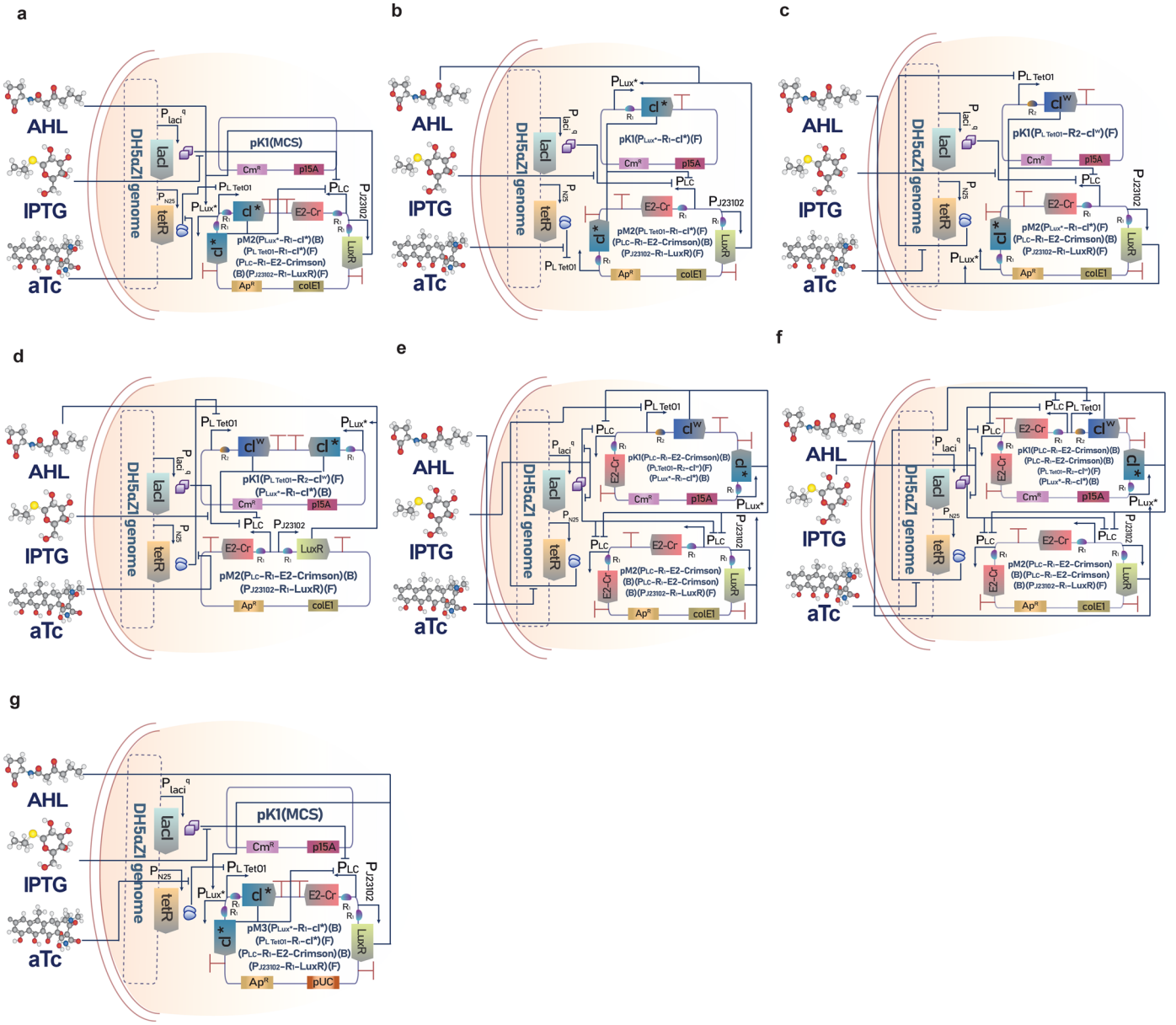
Extended Data Fig. 5 | Details of characterization and Dose Responses experiments across iterations of Bactoneuron 7. AS7A, AS7B, AS7C and AS7D; yielding corresponding parameter values (shown in figure) of Fold Change (F.C.), Σ Leakage (ΣL), Signal variation (S.V.), Bias ($B_{AS\#}$), Weights ($W_{Inducer}$) and Inducer Saturation points (Sat.). The fluorescence readings obtained from the dose response experiments were fitted with the Log-Sigmoid function (see Methods) to derive the Weight and Bias parameters. The induction states of the two constant inducers are specified atop the fitted curves. The final selected Construct is shown boxed. Each data point represent mean and s.d. from four independent colonies.

https://doi.org/10.1038/s41589-024-01711-4



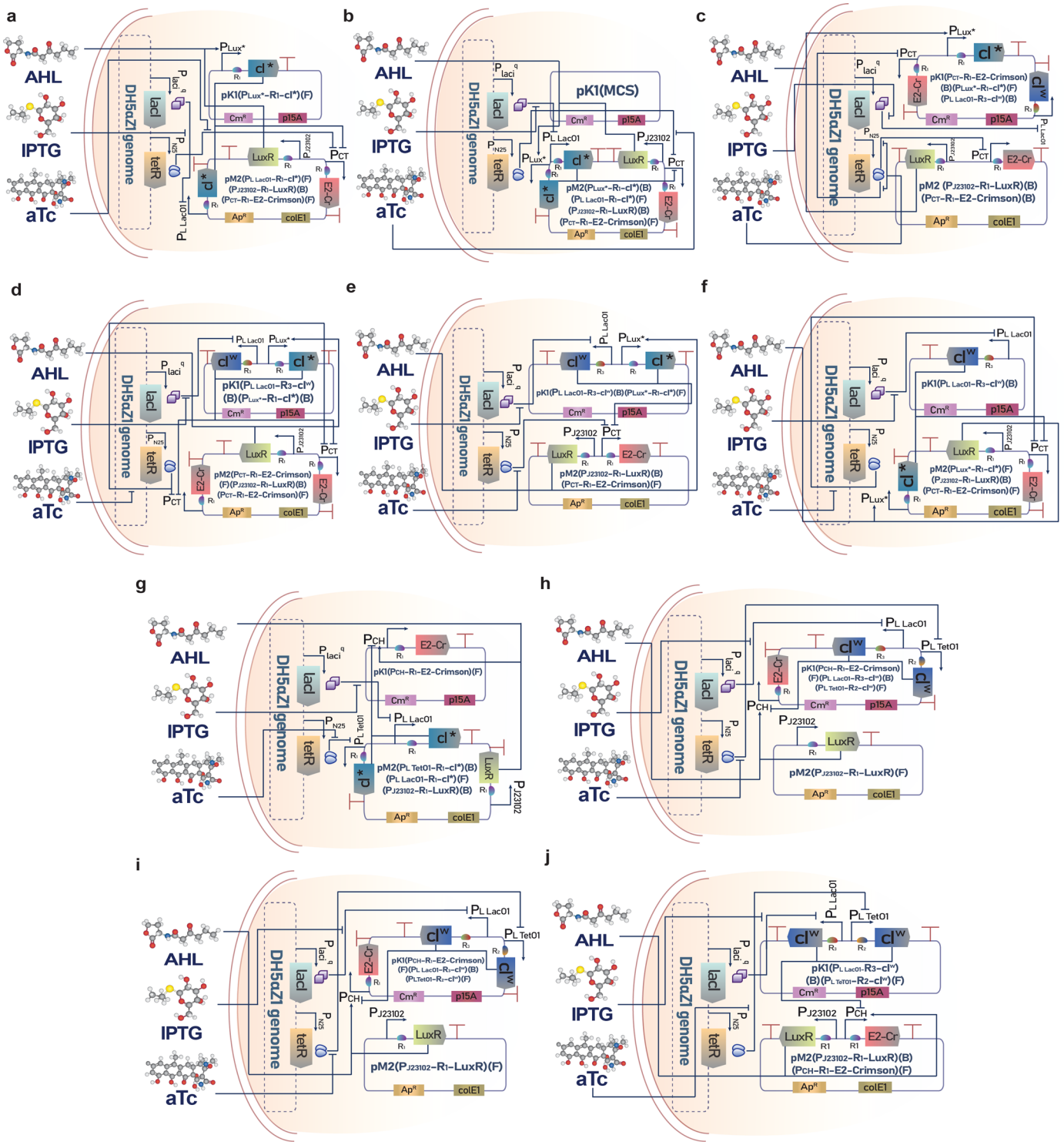
Extended Data Fig. 6 | Detailed genetic circuit designs and plasmid maps of Bactoneurons that went through no iteration and were selected as final. (**a**) BNeu AS1, (**b**) BNeu AS2, (**c**) BNeu AS3, (**d**) BNeu AS4, (**e**) BNeu AS8, (**f**) BNeu AS9. Names of plasmids constructed and incorporated in the circuits are also shown within the plasmid schematic. MCS represents empty network brick cloning site. 'F' and 'B' denotes forward and reverse direction of cassette respectively. The E2-crimson output of AS1 is altered with EGFP to give AS1* and EGFP output of AS2 is altered with E2-Crimson to give AS2*.

https://doi.org/10.1038/s41589-024-01711-4



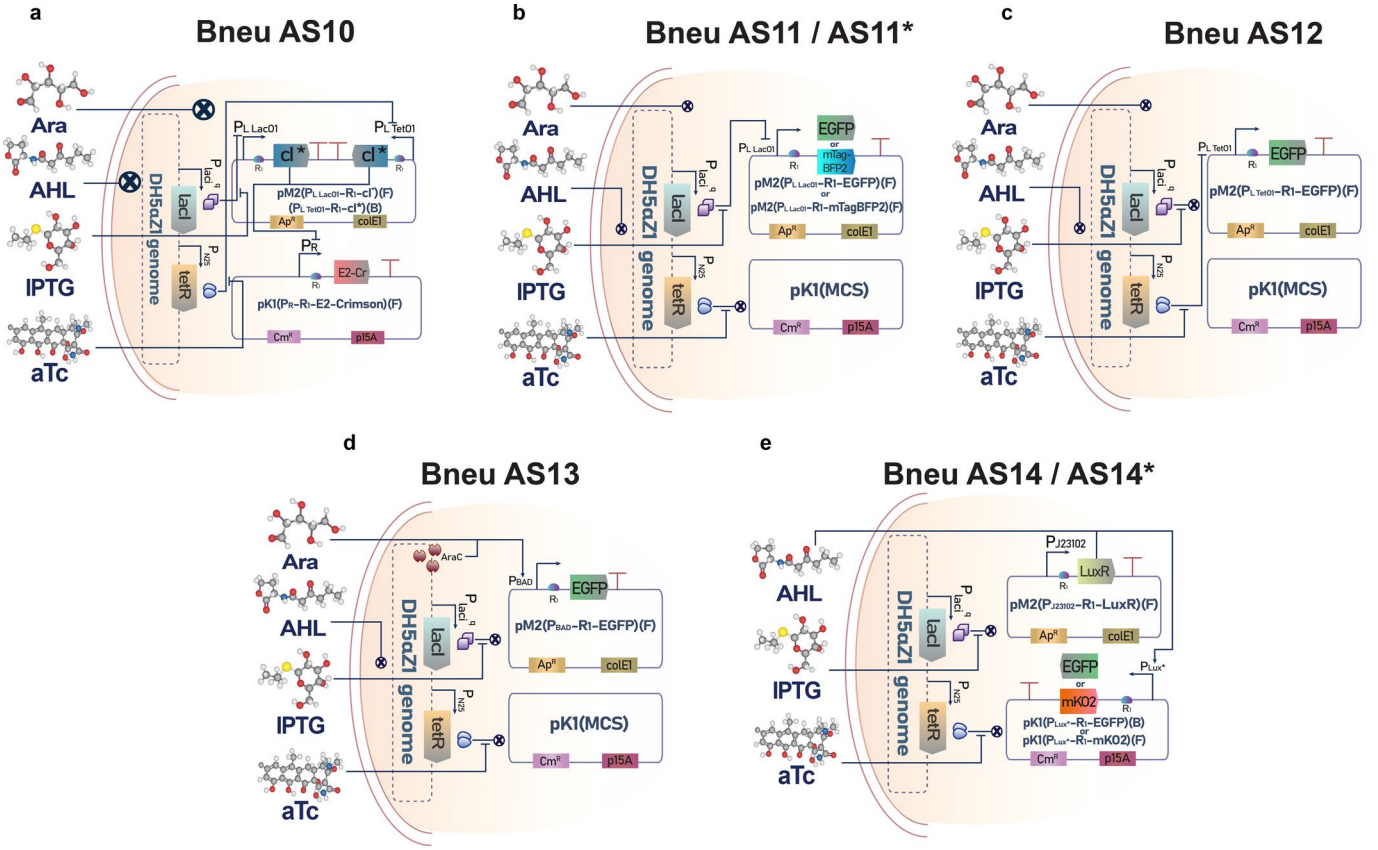
Extended Data Fig. 7 | Detailed genetic circuit designs and plasmid maps of Bactoneuron 5 across iterations. (**a**) BNeu AS5A, (**b**) BNeu AS5B, (**c**) BNeu AS5C, (**d**) BNeu AS5D, (**e**) BNeu AS5E, (**f**) BNeu AS5F and (**g**) BNeu AS5G. Names of plasmids constructed and incorporated in the circuits are also shown within the plasmid schematic. MCS represents empty network brick cloning site. 'F' and 'B' denotes forward and reverse direction of cassette respectively.

https://doi.org/10.1038/s41589-024-01711-4



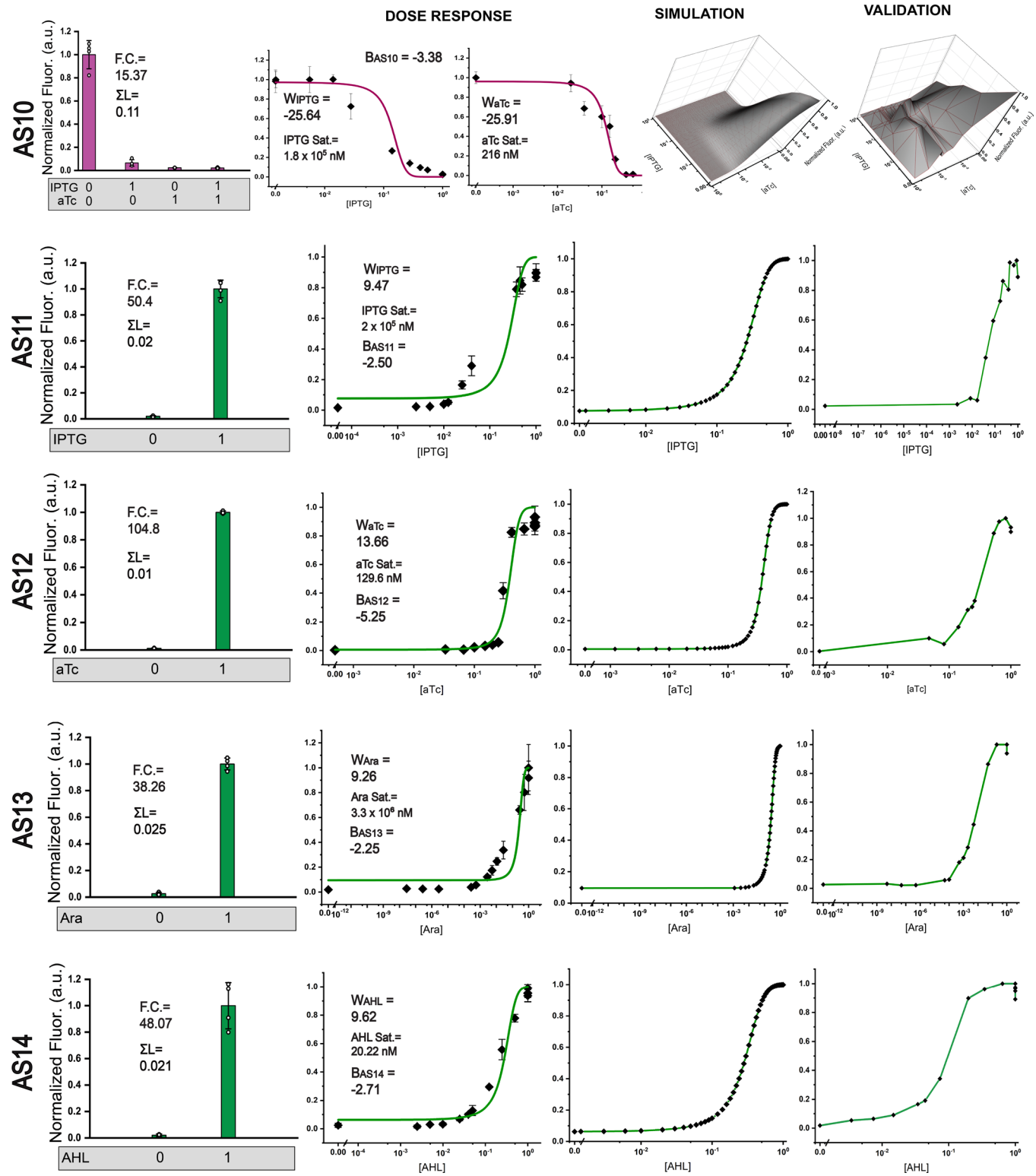
Extended Data Fig. 8 | Detailed genetic circuit designs and plasmid maps of Bactoneuron 6 and Bactoneuron 7 across iterations. (**a**) BNeu AS6A, (**b**) BNeu AS6B, (**c**) BNeu AS6C, (**d**) BNeu AS6D, (**e**) BNeu AS6E, (**f**) BNeu AS6F, (**g**) BNeu AS7A, (**h**) BNeu AS7B, (**i**) BNeu AS7C and (**j**) BNeu AS7D. Names of plasmids constructed and incorporated in the circuits are also shown within the plasmid schematic. MCS represents empty network brick cloning site. 'F' and 'B' denotes forward and reverse direction of cassette respectively.

https://doi.org/10.1038/s41589-024-01711-4



Extended Data Fig. 9 | Detailed genetic circuit designs and plasmid maps of Bactoneurons. (**a**) BNeu AS10, (**b**) BNeu AS11 and AS11*, (**c**) BNeu AS12, (**d**) BNeu AS13, (**e**) BNeu AS14 and AS14*. Names of plasmids constructed and incorporated in the circuits are also shown within the plasmid schematic. MCS represents empty network brick cloning site. 'F' and 'B' denotes forward and reverse direction of cassette respectively. The EGFP output of AS11 is altered with mTagBFP2 to give AS11* and EGFP output of AS14 is altered with mKO2 to give AS14*.

https://doi.org/10.1038/s41589-024-01711-4



Extended Data Fig. 10 | Experimental Characterizations of Bactoneurons. Fold change characterization, Dose Responses experiments with fitting values, simulation, and re-validation of Bactoneurons AS10, AS11, AS12, AS13, and AS14. Each data point represent mean and s.d. from four independent colonies.

nature portfolio

Corresponding author(s): Sangram Bagh

Last updated by author(s): May 30, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Fluorescence and optical density measurement data were collected using Synergy HTX Multi-Mode reader and Gen5 v2.09 software (BioTek). Primers were designed using Primer 3 (version 4.0). RBS strength was calculated using RBS Calculator (version 2.1). Microscopy data was obtained from Nikon A1R-Si confocal microscope and EVOS M5000 Cell Imaging System (Thermo Fisher Scientific) using NIS-Elements AR and EVOS 2v1.6.1899 softwares respectively.
Data analysis	Microsoft Excel (16.0.1753), OriginPro 2018, ImageJ (Fiji) softwares were used for analysis of the data.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All relevant data supporting the findings are available within the paper, Extended Data and the Supplementary Data. Source data are provided with the manuscript.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Experiments were performed in quadruplicates, where four independent agar plate colonies were inoculated to liquid culture, in order to observe differences in controlled gene expression behavior.
Data exclusions	None
Replication	Successful attempts to replicate the results have been observed using freshly prepared , media, reagents and cells. Every modular cells have been used among many multicellular systems and same cellular modules were used multiple times in various experiments repeatedly over the past three years.
Randomization	Randomization was achieved by streaking freshly prepared agar plates with freshly transfected engineered bacterial cells , followed by the selection of individual colonies from subsequent plates randomly. This process ensured a randomized sampling of each tested engineered cells.
Blinding	The study is not blinded as the experiments and analysis were performed by the same group of researchers.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging